VŠB TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Tvorba webové aplikace pro vizualizaci měřených dat za použití platformy LabVIEW NXG Web Modul

Creation of Web Application for Visualization of Measured Data using LabVIEW NXG Web Module

Jan Maléř

Bakalářská práce

Vedoucí práce: Ing. Pavel Kodytek

Ostrava, 2021

Abstrakt

Bakalářská práce se zabývá vytvořením webové aplikace, která dokáže komunikovat se standardním vývojovým prostředím LabVIEW a následnou vizualizací měřených dat v aplikaci. Je v ní představeno vývojové prostředí, jeho vlastnosti a součásti, a také základní způsob tvorby webových stránek. Následně popisuje princip komunikace webové a měřící aplikace s centrálním serverem. V následující kapitole se zaměřuje na vstupní data aplikace a jejich simulaci. V závěru práce prezentuje webovou aplikace a vizualizaci měřených dat.

Klíčová slova

webová aplikace; LabVIEW; LabVIEW NXG; HTML; CSS; JavaScript; Tag; SystemLink; komunikace; internet; vizualizace

Abstract

This bachelor's thesis deals with the creation of a web application that can communicate with the standard development environment LabVIEW and the subsequent visualization of measured data in the application. There is introduced the development environment, its features and components, as well as basic process of creating websites. Describes the principle of communication of web and measuring applications with central server. In next chapter focuses on the input data of the application and their simulation. In conclusion presents the web application created and the visualization of measured data.

Keywords

Web application; LabVIEW; LabVIEW NXG; HTML; CSS; JavaScript; Tag; SystemLink; Communication; Internet; Visualization

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Pavlu Kodytkovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam použitých zkratek a symbolů

ADC	–	**A**nalog-to-**D**igital **C**onverter (analogově digitální převodník)
API	–	**A**pplication **P**rogramming **I**nterface (rozhraní pro programování aplikací)
CSS	–	**C**ascading **S**tyle **S**heets (kaskádové styly)
DBL	–	**D**ou**BL**e (datový typ)
HTML	–	**H**yper**T**ext **M**arkup **L**anguage (hypertextový značkovací jazyk)
LabVIEW	–	**Lab**oratory **V**irtual **I**nstrument **E**ngeneering **W**orkbench
OPC	–	**O**LE for **P**rocess **C**ontrol
SGML	–	**S**tandard **G**eneralized **M**arkup **L**anguage (standardní jazyk pro popis značkovacích programů)
URL	–	**U**niform **R**esource **L**ocator (jednotný lokátor zdroje)
VI	–	**V**irtual **I**nstrument (virtuální přístroj)
W3C	–	**W**orld **W**ide **W**eb **C**onsortium (Mezinárodní webové konsorcium)
WWW	–	**W**orld **W**ide **W**eb (celosvětová síť)
WYSIWYG	–	**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et (co vidíš, to dostaneš)
XHTML	–	e**X**tensible **H**yper**T**ext **M**arkup **L**anguage (rozšiřitelný hypertextový značkovací jazyk)
XML	–	e**X**tensible **M**arkup **L**anguage (rozšiřitelný značkovací jazyk)

Seznam obrázků

Seznam tabulek

Kapitola 1

Úvod

Tato bakalářská práce se věnuje především vytvoření webové aplikace a aplikace pro měření a sběr dat. S použitím nového programovacího prostředí LabVIEW NXG souvisí využití nových možností. Mezi tyto možnosti patří také tvorba webových aplikací. Hlavním úkolem je zajistit komunikaci se standardním vývojovým prostředím LabVIEW. Výhodou webové aplikace je přístup uživatele k datům měření odkudkoliv na světě. Konečným výstupem sloužícím jako demonstrace funkčnosti webové aplikace je ovládání a vizualizace dat měřící aplikace skrze internet. Součástí měřící aplikace je také simulace výstupních dat. Ta nahrazuje měřená data získaná praktickým měřením.

Samotná práce je logicky rozdělena do několika částí. První dvě kapitoly jsou věnovány seznámení se s vývojovým prostředím LabVIEW a jeho novějším následovníkem LabVIEW NXG. Součástí druhé kapitoly je rovněž popis pluginu, který je klíčovým prvkem celé práce s webovou aplikací. Navazuje kapitola, ve které je pojednáno o webových stránkách, jejich tvorbě a vizuálních úpravách. Následuje velmi důležitá část práce, kterou je vzájemná komunikace obou aplikací pomocí cloudového serveru. Ta je nezbytná pro správnou funkčnost obou aplikací. Poslední dvě kapitoly obsahují podrobný popis vzniklých aplikací, kterými jsou měřící a webová aplikace.

Kapitola 2

LabVIEW

LabVIEW™ je systémové a vývojové prostředí vyvinuto v roce 1986 společností *National Instruments Corporation*, dnes již *NI*, sídlící v Austinu v Texasu. Tehdy vznikl pojem virtuální instrumentace tzn. virtuální přístroje se všemi náležitostmi hardwarového přístroje. Cílem LabVIEW je usnadnit vědcům a technikům, jež nejsou programátoři, vytvoření jejich testovacích a měřicích systémů. Předešlé zkušenosti v oblastech vývoje softwaru kontrolních přístrojů, vedly vývojáře k tomu, aby bylo LabVIEW modelováno jako hierarchie virtuálních přístrojů. Virtuální přístroje na nejnižší úrovni jsou jednoduše odrazy jednotlivých fyzických přístrojů, které vědci a technici uměli ovládat. Virtuální přístroje vyšší úrovně kombinují přístroje nižší úrovně, aby mohly poskytovat složitější měření. [1, 2, 3, 4]

2.1 Vlastnosti LabVIEW

LabVIEW slouží pro sběr a zpracování naměřených analogových a digitálních dat. Hlavní vlastností odlišující LabVIEW od ostatních programů pro sběr dat, je především jeho vysoce modulární grafický programovací jazyk nesoucí název *G*. Tedy místo psaní příkazů se sestavují blokové diagramy, schémata datových toků. Program se nazývá VI – **V**irtual **I**nstrument.

LabVIEW se obvykle používá v aplikacích pro testování a měření, které obsahují hardware (jako je NI CompactDAQ), ale není omezen na hardware NI. LabVIEW má také obsáhlou knihovnu matematických a statistických funkcí a konstant. Ty jsou interpretovány ikonami, které lze vzájemně spojovat virtuálními vodiči a výsledky pak vykreslovat např. do grafů. Každý prvek má příslušný datový typ a ty jsou přehledně odlišeny svou barvou. Vodiče spojující prvky přizpůsobí svůj styl, barvu a hrubost čáry vzhledem k datovému typu. Výhodou grafického programování je intuitivní ovládání jako u fyzických přístrojů a flexibilní, znovu použitelný kód. Podprogramy mohou být uloženy do knihoven a použity bez nutnosti modifikace v jiných programech, což značně zkracuje dobu vývoje. Řízení chodu programu je dáno tokem dat (data flow) neboli, že se daná část kódu

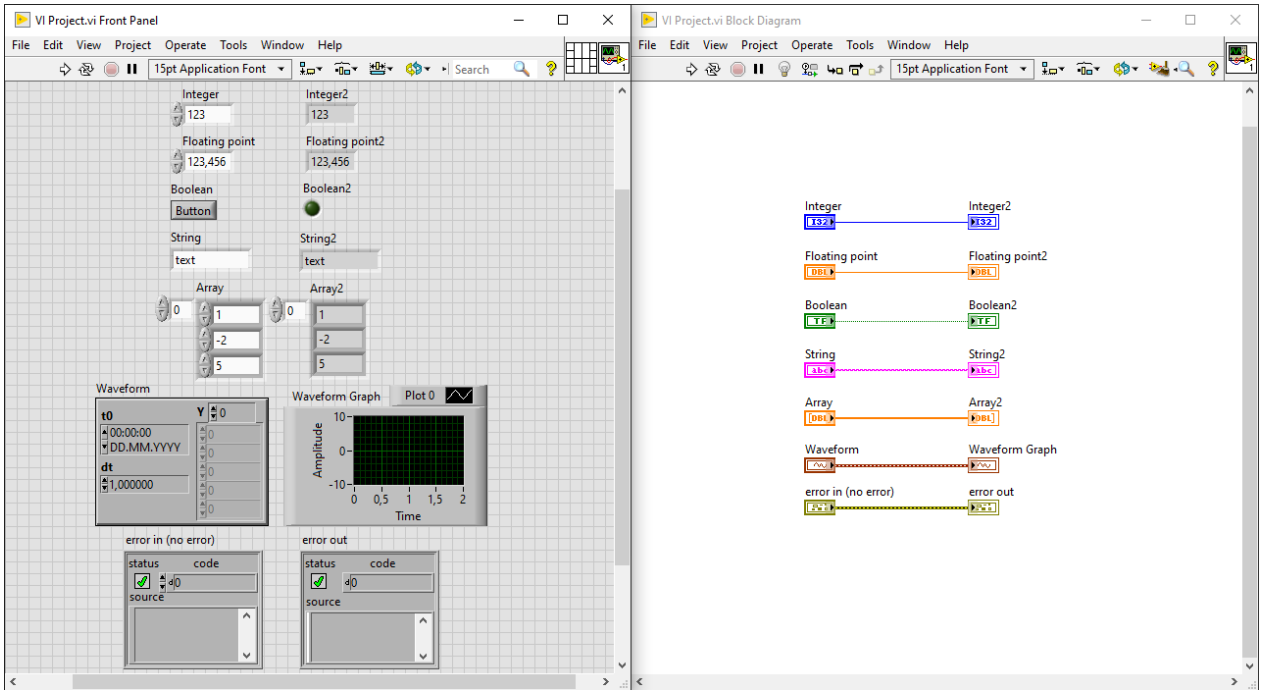
(uzel) vykoná až tehdy, když má na vstupech data z předchozích částí kódu (uzlů). LabVIEW lze použít k vytváření jednoduchých i složitých softwarových aplikací. [1, 2, 3, 4, 5]

2.2 Porovnání jazyka G s jazykem C

Programovací jazyk C je optimalizován pro sekvenční provádění instrukcí rychlostí, závisející na rychlosti CPU. Tzn. program je vykonáván sérií po sobě jdoucích instrukcí. Naopak grafická syntaxe v LabVIEW je optimalizována pro paralelní provádění instrukcí. C je nízko úrovňový, tedy je nutno zvážit a určit i ty nejmenší podrobnosti, jako jsou přiřazení paměti a vlákna. Díky tomu může programátor optimalizovat vyvíjenou aplikaci, tedy využít výhod architektury operačního systému, na který je aplikace vyvíjena a tím dosáhnout vyššího výkonu aplikace. Z tohoto hlediska je většina knihoven v LabVIEW napsána v jazyce C/C++. Díky tomu jsou operace (např. analýza dat) v LabVIEW stejně rychlé jako v C, protože jsou optimalizovány pro každou platformu a operační systém, které LabVIEW podporuje. Uživatel pak dále píše v jazyce G, který tyto knihovny používá, a nemusí se zabývat např. přiřazením paměti a vlákna. [6]

2.3 Vývojové prostředí

Vývojové prostředí LabVIEW se skládá ze dvou vzájemně propojených panelů:



Obrázek 2.1: Čelní panel a blokový diagram s ukázkou datových typů v LabVIEW 2020

- **Čelní panel (Front panel)** je uživatelské rozhraní VI, umisťují se zde ovládací prvky, indikační prvky a dekorační prvky.
  - **Ovládací prvky**: Tlačítka, přepínače, posuvné stupnice a další prvky pro zadávání dat. Mají stejný účel jako ovládací prvky fyzických měřicích přístrojů a předávají data do blokového diagramu.
  - **Indikační prvky**: Grafy, LED, teploměry, ručičkové nebo digitální ukazatele a další. Jako u fyzických přístrojů jsou indikační prvky interpretací měřených nebo generovaných dat v blokovém diagramu.
  - **Dekorační prvky**: Nejedná se o vstupy ani o výstupy. Jejich účelem je graficky doplnit čelní panel, aby byla aplikace přehledná a dobře se s ní pracovalo.
- **Blokový diagram (Block diagram)** obsahuje grafický zdrojový kód, který po spuštění definuje posloupnost vyhodnocení jednotlivých složek VI. Ovládací prvky a indikátory se po přidání v čelním panelu automaticky umístí do blokovém diagramu, zobrazují jako terminály.
  - **Terminály**: Vstupní a výstupní body ovládacího prvku nebo indikátoru. Terminály mohou být zobrazeny pomocí ikony nebo standardně (bez ikony). Datový typ terminálu je reprezentován určitou barvou a popisem uvnitř ikony. Vstupní terminály jsou ohraničeny silnější čarou s malou šipkou napravo směřující ven z terminálu. Výstupní terminály jsou ohraničeny slabší čarou s šipkou nalevo směřující dovnitř terminálu.
  - **Uzly**: Objekty blokového diagramu se vstupy a výstupy. Vykonávají operace při běhu VI. Jsou obdobou rozhodovacích výrazů, operátorů, funkcí a podprogramů v textově orientovaných programovacích jazycích.
  - **Vodiče**: Přenos dat mezi uzly je zprostředkován vodiči. Každý vodič má jeden zdroj dat, ale může být použit do mnoha VI a funkcí, které čtou data. V závislosti na datovém typu mají vodiče různé barvy, styly a tloušťky. Neplatný vodič je černý přerušovaný s červeným křížkem.
  - **Struktury**: Grafické znázornění smyček a rozhodovacích výrazů textově orientovaných programovacích jazyků. [4, 7, 8]

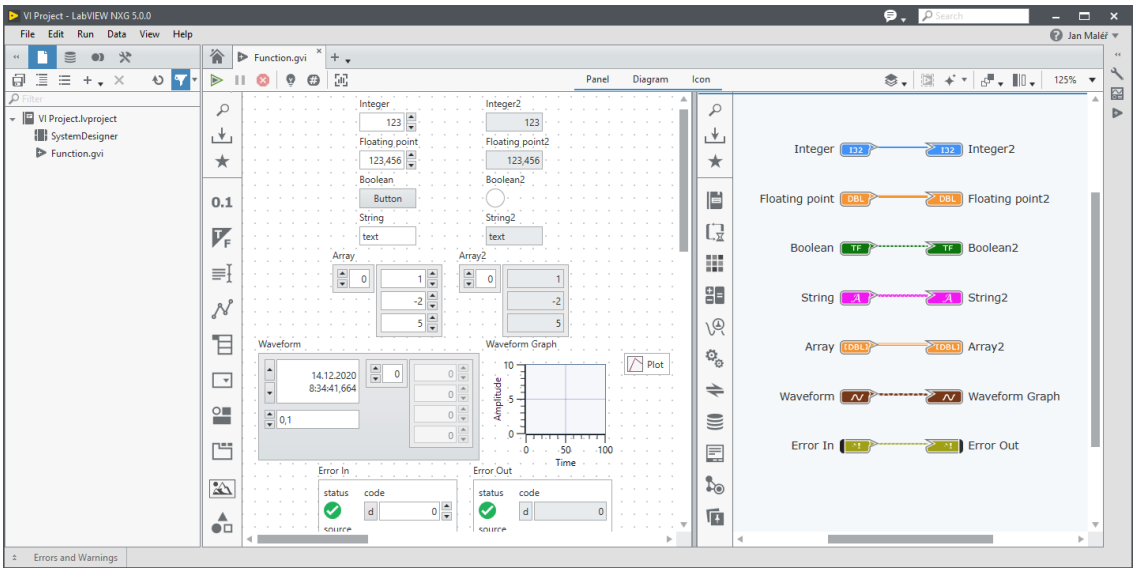
Kapitola 3

LabVIEW NXG

V roce 2017 na konferenci NI WEEK, představilo NI spolu s novou verzí LabVIEW 2017 také jeho novou generaci: **LabVIEW NXG**. Jedním z hlavních cílů vytvoření nové generace bylo zlepšit uživatelům komfort při práci, zejména novým uživatelům poskytnou modernější vývojové prostředí se snadno dostupnými prvky a stávajícím uživatelům nabídnout usnadnění při tvorbě nových projektů. LabVIEW NXG dokáže automaticky rozpoznat a podporovat konfiguraci CompactRIO, včetně různých nainstalovaných modulů, což výrazně usnadňuje vytváření a konfiguraci projektů. Prostředí taktéž automaticky stáhne a nainstaluje potřebné softwarové ovladače pro prvky deklarované v projektu. Každý modul lze navíc okamžitě konfigurovat z uživatelského rozhraní. LabVIEW NXG v zásadě vytváří malý program nebo virtuální přístroj pro každé zařízení, například analogově-digitální převodník (ADC). VI může mít libovolný počet konfiguračních parametrů, které lze okamžitě upravit. Nakonfigurované VI lze poté zkopírovat a vložit do projektu, kde jej lze propojit spolu s dalšími VI. Schopnost rychlé konfigurace systému šetří čas při spuštění a novým uživatelům usnadní začátky. LabVIEW NXG používá stejný grafický programovací jazyk G a jeho kompilátor jako LabVIEW. To umožní případné sloučení obou vývojových platforem v budoucnu. [5, 9]

3.1 Vývojové prostředí

Vývojové prostředí v LabVIEW NXG je kompletně předěláno a navrženo tak, aby bylo přívětivé pro nové uživatele, kteří se s LabVIEW dosud nesetkali a nebyli ovlivněni zásadami z jiných verzí LabVIEW. Vzhledem k dnešnímu trendu jsou oproti klasickému LabVIEW všechny záložky a ovládací prvky situovány v jednom okně.



Obrázek 3.1: Čelní panel a blokový diagram s ukázkou datových typů v LabVIEW NXG

3.2 Web Module

Webový modul umožňuje dálkově ovládat a sledovat testovací měřicí systémy. Dokáže vytvářet webové stránky přímo v LabVIEW a usnadňuje tak přechod mezi VI a webem. Je to možné díky novým LabVIEW VI – webVI. WebVI se skládá ze standardních webových technologií jako jsou HTML, CSS a JavaScript. To jim umožňuje spustitelnost v kterémkoliv webovém prohlížeči bez dalších plug-inů a jiných aplikací. Součástí webového modulu jsou také prostředky pro komunikaci mezi dálkovými měřicími systémy a uživatelským prostředím. Uživatelské prostředí vytvářené na čelním panelu je automaticky generováno do HTML, CSS a JavaScriptu. Tento kód lze upravovat, přidávat součásti z jiných zdrojů a tím ovlivnit celé uživatelské prostředí. Pro různé typy zařízení (PC, tablet, mobil) lze upravit velikost uživatelského prostředí a vyhnout se tak rozdílům velikostí různých platforem. V blokovém diagramu kromě programu lze vytvořit komunikaci mezi testovacími měřicími systémy a uživatelským prostředím. Tuto komunikaci lze realizovat přes HTTP nebo pomocí již zabudovaných SystemLink Tag API a SystemLink Message API. Testování a ladění může být prováděno buď lokálně nebo nahráním na NI Web Server. NI Web Server poskytuje všechny průmyslové standardní způsoby zabezpečení. [10]

3.3 SystemLink

SystemLink™ je platforma funkcí zaměřená na optimalizaci provozní efektivity. Poskytuje propojenou inteligenci pro automatizované testovací a automatizované měřicí systémy. Disponuje centrální webovou aplikací, která zvyšuje efektivitu úloh vzdálené konfigurace a procesů nasazení softwaru.

SystemLink je navržen speciálně pro zlepšení konfiguračních procesů pro velké množství distribuovaných testovacích a měřicích systémů, dále pro zlepšení instalace softwaru, aktualizace zařízení, diagnostické funkce, monitorování výsledků testů a správu dat měření. Přestože je SystemLink v souladu s testovacími a měřicími řešeními založenými na LabVIEW, nabízí také otevřenou architekturu, která podporuje software a hardware třetích stran.

SystemLink poskytuje aplikace a nástroje, které zlepšují provozuschopnost systému a monitorují výkonnost měřicích a testovacích přístrojů. Nepřetržité monitorování stavu systému pomocí předkonfigurovaných monitorovacích služeb a výkonu systému je nutné pro správný chod výroby. Zjištění problémů již na počátku lze zabránit tomu, aby se drobné problémy systému staly závažnými poruchami v provozu. Je možno e-mailem upozornit připojené uživatele pomocí konfigurovatelného alarmového modulu s již uživatelem nakonfigurovaných e-mailových upozornění. [11, 12, 13]

3.3.1 Součásti architektury SystemLink

- **SystemLink Web Application**: Skrze webové rozhraní v prohlížeči lze provádět úkony jako je vzdálená administrace systémů, konfigurace alarmů, plánování zpracování dat, monitorování testů a následné hlášení jejich výsledků. Webová aplikace běží na počítači, na kterém je současně spuštěn SystemLink Server a již zahrnutý NI Web Server.

- **SystemLink Server**: Osahuje softwarové programy od NI a technologické služby a API rozhraní na straně serveru běžící na počítači, který představuje hlavní uzel v komunikaci aplikace a dat.

- **NI Web Server**: Poskytuje hostitele aplikačního serveru pro SystemLink Web Application. Je součástí instalace SystemLink Severu. Tento server využívá také LabVIEW NXG Web Module. Umožňuje konfigurovat serverové nastavení uživatelských rolí a oprávnění, ověřování uživatelů a zabezpečení.

- **SystemLink Client**: Softwarový program, je instalován na počítače, na kterých je potřeba distribuované výpočetní prostředí SystemLink. Klient zajišťuje připojení k SystemLink Serveru a provádí příkazy na základě instrukcí ze serveru, buď z webové aplikace nebo z příkazů API. Mezi funkce klienta patří nasazení systému, monitorování testů a následné hlášení jejich výsledků. [13]

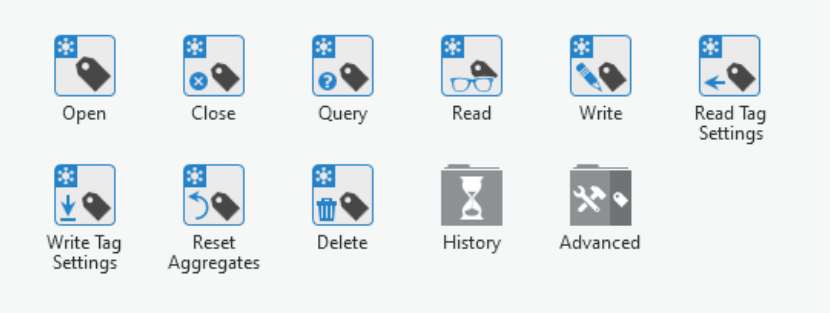
3.4 SystemLink Cloud

SystemLink Cloud je služba hostovaná v síti NI v zabezpečeném, škálovatelném cloudovém výpočetním prostředí s řídícím panelem a službou pro ukládání dat, díky kterému odpadá potřeba dalšího hardwaru pro správu infrastruktury a konfigurace serveru a zabezpečení webového serveru. V tom tkví výhody pro vývojové týmy, aby mohly snadno sdílet svá měřená data a poznatky s odběrateli

globálním a mobilním bezpečným způsobem. Umožňuje bezpečně přistupovat, monitorovat a komunikovat s aplikacemi odkudkoli na světě bez zátěže hostování a správy serveru na vlastním zařízení. [14, 15]

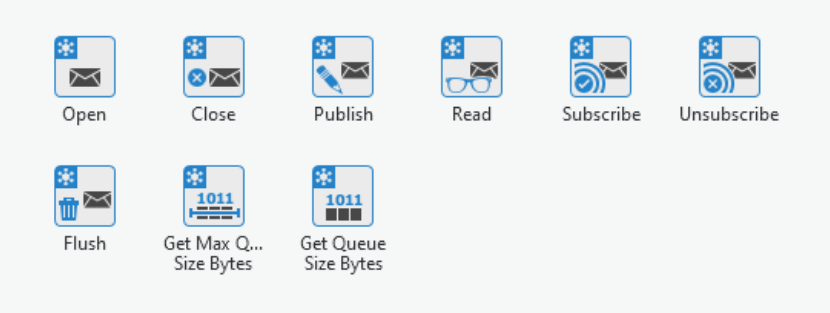
3.5 API

- **Tag API**: Tagy se používají k odesílání malého objemu občasných dat mezi distribuovanými systémy a aplikacemi. Nejsou vhodné pro přenos vysokorychlostních dat měření. Jsou vysoce škálovatelné a SystemLink Cloud se s nimi rychle synchronizuje. Pomocí tagů lze odesílat a přijímat data o testování a měření ze systému do jiných systémů, buď SystemLink Server nebo SystemLink Cloud. [13, 14]



Obrázek 3.2: Nabídka Tagů v LabVIEW NXG

- **Message API**: Zprávy umožňují komunikaci mezi systémy pomocí modelu. Pomocí tohoto rozhraní lze odesílat varovné zprávy, aktualizace stavu nebo příkazy ze systému do jiných systémů. Pro přijímání zpráv lze použít jakýkoli uzel ve spravovaném systému po přihlášení se k odběru. Dále může toto rozhraní přenášet velkoobjemová a vysokorychlostní data, mimo jiné i video. Dostatečná rychlost sítě a vhodný design aplikace jsou důležitými vlastnostmi pro správné zpracování a vizualizaci zpráv. [13, 14]



Obrázek 3.3: Nabídka Messages v LabVIEW NXG

Kapitola 4

Webová stránka

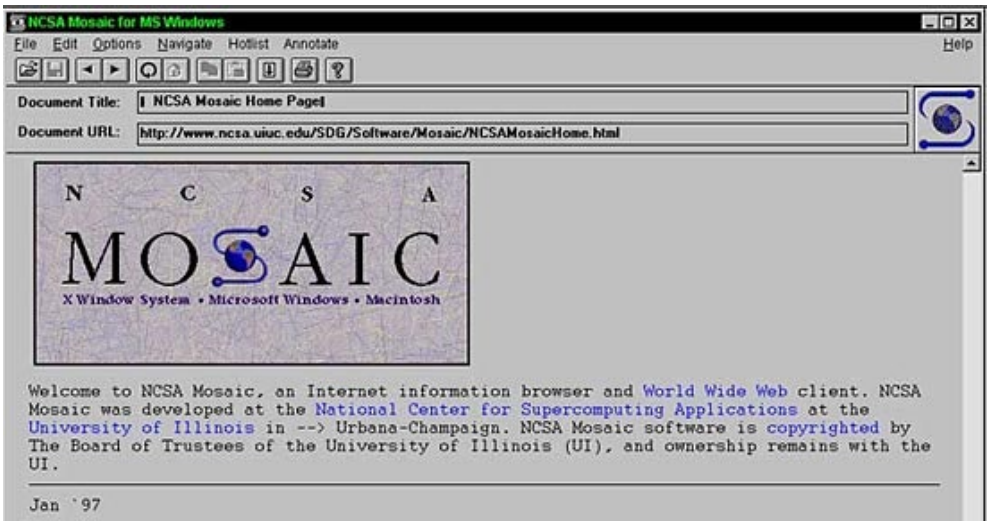
Webová stránka je dokument dostupný na World Wide Webu (WWW). Webové stránky jsou uloženy na webovém serveru, stránky lze zobrazit pomocí některého z webových prohlížečů. Stránka může obsahovat veliké množství informací včetně textu, grafiky, zvuku, videa a hypertextových odkazů. Tyto hypertextové odkazy jsou odkazem na jiné webové stránky. Každá webová stránka má přiřazenu jedinečnou adresu URL. Webové stránky se dále dělí na statické a dynamické.

Statické stránky jsou do prohlížeče klienta nahrány přesně tak, jak jsou uloženy na webovém serveru. Obsahují pouze statické informace, které může uživatel pouze číst, nemůže je jakkoli upravovat ani s nimi pracovat. Takové stránky jsou vytvářeny pouze pomocí HTML.

Dynamické stránky zobrazují různé informace v různém čase. Bez načtení celé webové stránky lze měnit její vzhled. Toho lze docílit skriptováním na straně serveru nebo na straně klienta. Skriptování na straně severu funguje jako rozhraní klienta a omezuje přístup uživatele k prostředkům na serveru. Skripty na straně klienta obsahují pokyny pro prohlížeč, které mají být provedeny v závislosti na akci uživatele. [16]

4.1 HTML

HTML je standardní značkovací jazyk pro vytváření webových stránek. Nejedná se o programovací jazyk, tedy nemá schopnost vytvářet dynamické funkce, ale umožňuje organizaci a formátování dokumentů podobně jako WYSIWYG textových editorů. Skládá se z řady prvků, které říkají webovému prohlížeči, jak má zobrazovat obsah stránky. V roce 1989 vzniklo HTML, když Sir Tim Berners-Lee přišel s návrhem použít hypertext k propojení souvisejících dokumentů v síti. Hypertext představuje text, který obsahuje odkaz na jiné texty, na které je možné se ihned dostat. V roce 1991 vyšla první verze HTML+ s 18 značkami (tzv. tagy), které bylo založeno na již používaném značkovacím jazyce SGML. Do roku 1993 byl dostupný pouze jeden textově založený prohlížeč. S příchodem nového prohlížeče Mosaic bylo již možné používat obrázky, vnořené seznamy a formuláře. [17, 18]



Obrázek 4.1: Webový prohlížeč Mosaic z roku 1993 [19]

O rok později byla založena skupina vývojářů pro vývoj specifikace HTML, která vzápětí vydala verzi HTML2. Specifikace HTML jsou udržovány a vyvíjeny konsorciem W3C (World Wide Web Consortium). S každou novou verzí rostl také počet nových tagů. Poslední verzí měla být HTML4.01, která vyšla v roce 1999, protože W3C se rozhodlo ubírat jiným směrem. Počátkem nového tisíciletí přišlo W3C s novou verzí lišící se od předchozích verzí pod názvem XHTML. Vycházelo z verze HTML4.01, která byla přepsána jako XML specifikace, což znamená, že všechny tagy musí být ukončeny, musí být psány malými písmeny, všechny atributy musí být uzavřeny v uvozovkách a tagy musí být vnořeny bez překrývání. Když jsou tyto podmínky splněny, soubor je validní a připraven pro zobrazení v prohlížeči. V opačném případě soubor nebude načten.

Díky těmto striktním požadavkům je psaní XHTML velmi obtížné. Většina webových designérů uznává důležitost vytváření validního HTML. Avšak to, že stránka není validní, neznamená, že ji prohlížeč nezobrazí. Nesprávně napsané části kódu se buď špatně zobrazí nebo nezobrazí vůbec. To vedlo webové návrháře a vývojáře k tomu, začít pracovat na specifikaci HTML5. V roce 2008 bylo rozhodnuto zrušit vývoj XHTML ve prospěch HTML5, které se stalo nejnovějším standardem. V roce 2017 byla vydána nejnovější verze HTML5.2. [17, 18]

4.2 Základní struktura HTML

HTML kód se skládá z elementů, ty většinou tvoří otevírací <tag> a ukončovací </tag>. Elementy předávají prohlížeči informace, které se nacházejí mezi otevíracím a uzavíracím tagem. Pokud na úplný začátek souboru napíšeme **<!DOCTYPE html>**, říkáme tím, že se jedná o HTML5. Každý HTML5 soubor musí vždy obsahovat tyto tři základní elementy:

1. **<html></html>**: Jedná se o první element dokumentu a zároveň jeho poslední. Vše, co se nachází mezi tagy je HTML kód. Můžeme v něm např. definovat jazyk dokumentu.

2. **<head></head>**: Hlavička obsahuje meta informace stránky pro prohlížeč, jako jsou např. informace o kódování a titulek stránky. Kód napsaný v této části se na stránce nezobrazí.

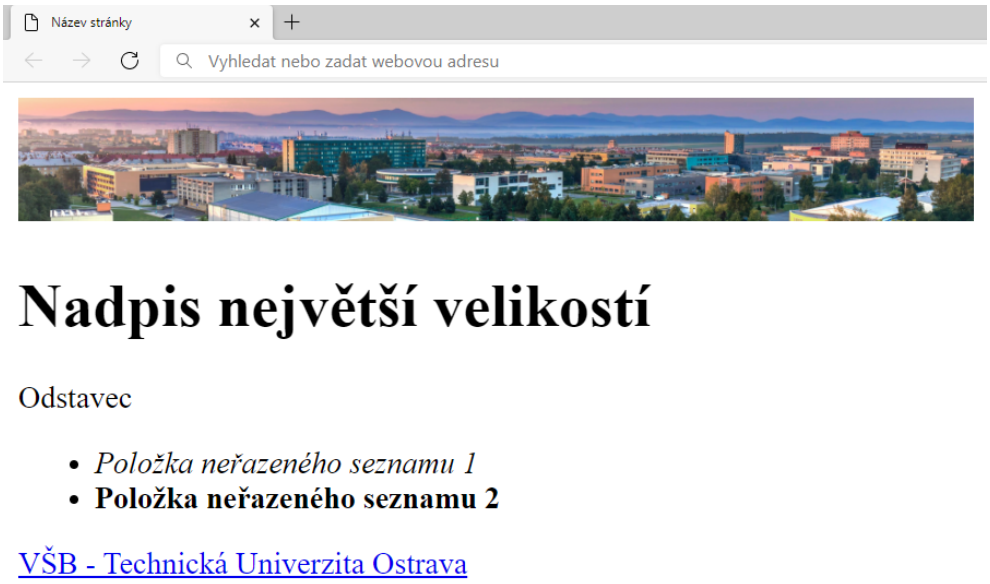
3. **<body></body>**: V těle se nachází obsah webové stránky, který se zobrazí na stránce v prohlížeči. Může obsahovat elementy jako jsou např. nadpisy, odstavce, odkazy, obrázky, tabulky atd. [20, 21]

4.2.1 Block-Level elementy

Block-Level elementy jsou takové elementy, které zabírají většinu dostupné plochy a vždy začínají na novém řádku. Mezi takové elementy se řadí např. nadpisy s 6 velikostmi **<h1>** až **<h6>**, odstavce **<p>**, oddíly ** **, seznamy **<ol>** nebo **<ul>** atd. Patří sem také výše zmíněné **<html>**, **<head>** a **<body>**. [20, 21]

4.2.2 Inline elementy

Inline elementy jsou takové elementy, které zabírají tolik místa, kolik vyžadují a nezačínají na novém řádku. Většinou se jedná o elementy upravující styl obsahu nacházejícího se v Block-Level elementech. Takovými elementy mohou být např. tučný text ****, kurzíva ****, podtržení **<u>**, zvýraznění **<mark>**, obrázky **<img>**, odkazy na jiné stránky **<a>** s atributem href definujícím adresu odkazu atd. [20, 21]



Obrázek 4.2: Ukázka jednoduché stránky v HTML

```
<!DOCTYPE html>
<html lang="cs-cz">
  <head>
    <meta charset="utf-8" />
    <title>Název stránky</title>
  </head>
  <body>
    <div>
      <img src="https://www.vsb.cz/cs/.content/galerie-obrazku/header_o-univerzite.jpg" width="500px">
      <h1>Nadpis největší velikostí</h1>
      <p>Odstavec</p>
      <ul>
        <li><em>Položka neřazeného seznamu 1</em></li>
        <li><strong>Položka neřazeného seznamu 2</strong></li>
      </ul>
      <a href="https://www.vsb.cz/cs/">VŠB – Technická Univerzita Ostrava</a>
    </div>
  </body>
</html>
```

Výpis 4.1: Kód ukázky jednoduché stránky v HTML

4.3 CSS

CSS jsou kaskádové styly definující, jak se mají HTML elementy zobrazovat na webové stránce. Namísto nastavování vzhledu u jednotlivých elementů opakovaně, CSS styly umožňují vytvořit globální pravidla určující vzhled elementu. Styl je konkrétnímu elementu přiřazen pomocí jednoznačného identifikátoru (id, class). Editací CSS lze snadno změnit vzhled webových stránek bez nutnosti zasahovat do HTML kódu. Kaskádové styly mohou být napsány buď přímo v HTML hlavičce nebo v souboru .css, které se připojí v hlavičce HTML souboru.

```
<html lang="cs-cz">
  <body>
    <p> <font color="blue"><b><u>
      "Lorem ipsum dolor sit amet, consectetur adipiscing  elit ,  sed do eiusmod tempor incididunt ut labore et
          dolore magna aliqua."
    </font></b></u></p>
    <p> <font color="blue"><b><u>
      "Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi  ut  aliquip  ex ea commodo
          consequat."
    </font></b></u></p>
  </body>
</html>
```

Výpis 4.2: Stylizace elementů v HTML bez CSS

```
<html lang="cs-cz">
  <head>
    <style>
      .modry {
        color: blue;
        font-weight: bold;
        text-decoration: underline;}
    </style>
  </head>
  <body>
    <p class="modry">"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
      incididunt ut labore et dolore magna aliqua."</p>
    <p class="modry">"Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
      commodo consequat."</p>
  </body>
</html>
```

Výpis 4.3: Stylizace elementů v HTML s CSS

V dnešní době se už webové stránky netvoří staticky, ale HTML kód je generován např. pomocí PHP, které umožňuje spojení s databází apod. V průběhu prohlížení již vygenerované webové stránky je možné kód modifikovat např. JavaScripty.

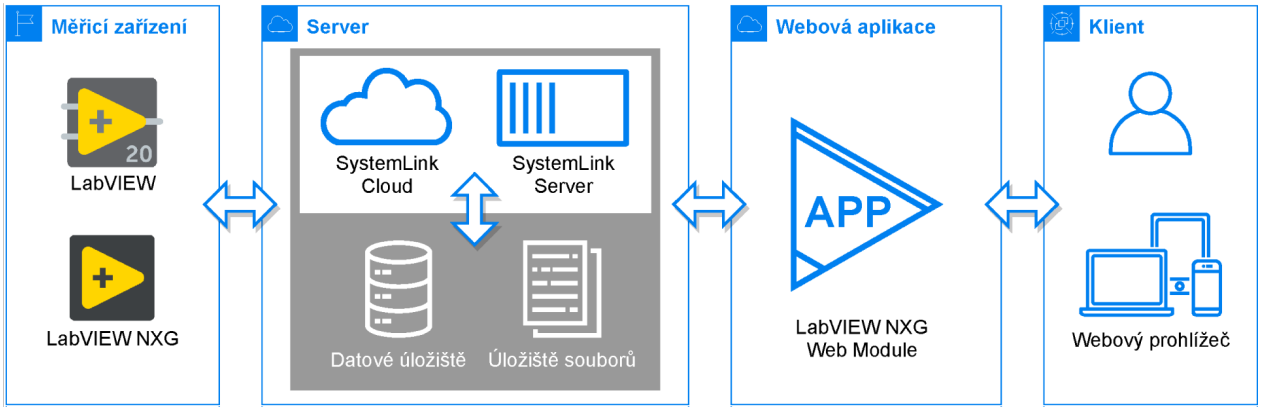
4.4 JavaScript

JavaScript patří k nejpopulárnějším programovacím jazykům. Jedná se o objektově orientovaný skriptovací jazyk patřící do rodiny jazyků C/C++/Java. Tento jazyk umožňuje do jinak statických webových stránek vkládat interaktivní prvky. Kód JavaScriptu není kompilován jako je tomu u jinačích jazyků. Je překládán po jednotlivých řádcích překladačem, který je zabudován v prohlížečích. JavaScript příkazy udávají prohlížeči, jakou akci má provést. Tyto příkazy jsou oddělovány středníkem. [22]

Kapitola 5

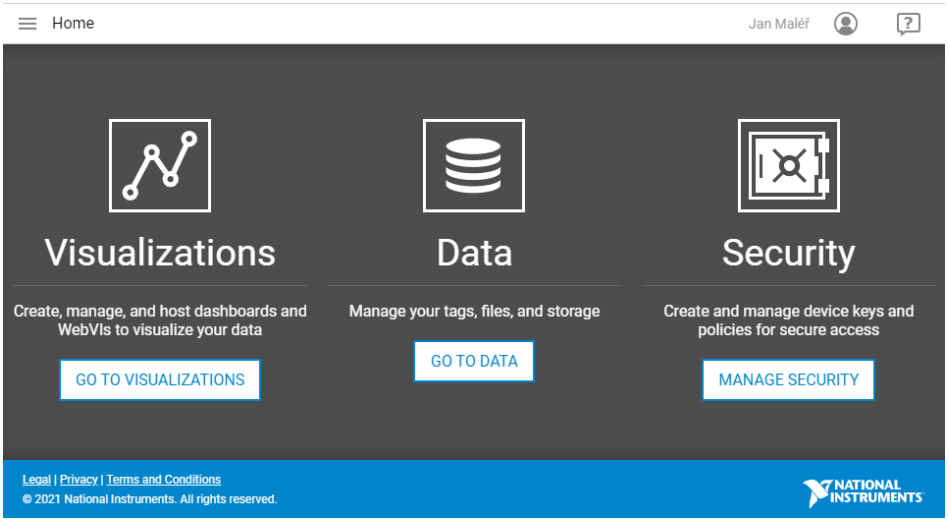
Komunikace

Webová aplikace je přístupový prvek, který může běžet na vzdáleném zařízení s webovým prohlížečem a přístupem k internetu. Uživatel se tak může připojit odkudkoli třeba na stolním počítači, notebooku nebo mobilu. Uživatel není limitován výkonem svého zařízení pro chod aplikace, ale jeho internetovým připojením. Díky tomu je uživatel schopen ovládat, případně získávat a číst data ze standardního prostředí LabVIEW.



Obrázek 5.1: Blokový diagram přenosu dat mezi aplikacemi a uživatelem

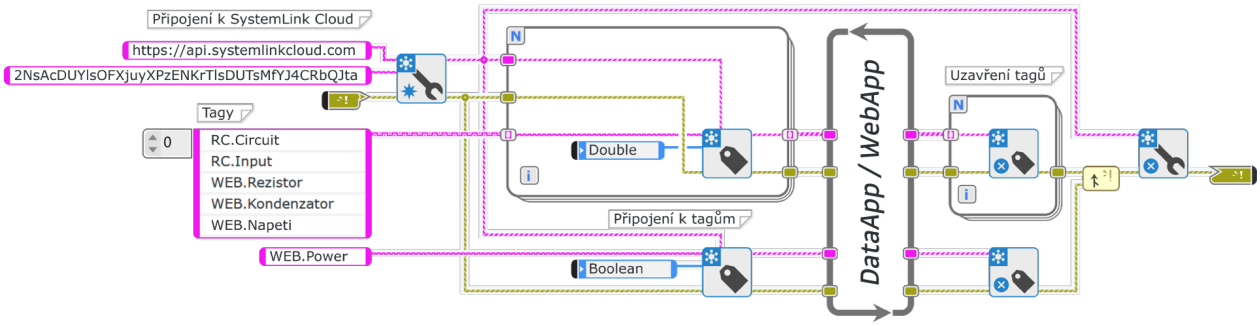
Taková webová aplikace může být kontrolním nebo i ovládacím prvkem k měřicí aplikaci. Ta je spuštěna v LabVIEW nebo LabVIEW NXG a posílá data do datového úložiště z místního počítače měřicí stanice. Aby bylo možné se k měřicí aplikaci připojit přes internet, musí být měřící i webová aplikace naprogramovány na stejný druh komunikace. SystemLink komunikace může být řešena Serverem nebo Cloudem (viz. kapitola 3.3). Připojení aplikací k těmto platformám se vůči sobě liší způsobem připojení tagů k programu. U připojení k SystemLink Serveru je zapotřebí nastavení adresy hostujícího serveru, uživatelské jméno a heslo. U SystemLink Cloudu je zadávána URL adresa Cloud serveru a API klíč (viz. obrázek 5.3). [23]



Obrázek 5.2: Hlavní stránka SystemLink Cloudu

5.1 Přopojení k SystemLink Cloud

V této práci je komunikace řešena pomocí Cloudu. API klíč je v SystemLink Cloudu generován a každý klíč je unikátní. V nastavení API klíče lze nastavit privilegia, která umožňují spravovat soubory a data. Data přicházející z měřicí aplikace se do datového úložiště přenášejí pomocí tagů. Ty umožňují přenos malého objemu dat mezi systémy a aplikacemi. Zjednodušeně je tato forma komunikace NI ekvivalentem OPC komunikace. Pomocí tagů je možno data ze systému do jiných systémů odesílat, ale také přijímat. Webová aplikace je zkompilována a následně nahrána na úložiště SystemLink Cloud. Aplikaci lze ihned po nahrání otevřít a začít používat. U aplikace lze opět nastavit privilegia a také její sdílení. Pokud je sdílení nastaveno na *Everyone*, aplikaci může ovládat kdokoli kdo má odkaz na aplikaci bez nutnosti NI účtu. Připojený uživatel má možnost měřit nebo ovládat měřicí přístroj připojený k webové aplikaci bez nutnosti být fyzicky přítomen u přístroje.



Obrázek 5.3: Blokový diagram připojení k SystemLink Cloudu

Na obrázku výše je vyobrazeno konkrétní připojení měřicí i webové aplikace k SystemLink Cloudu. Funkce se značkami klíče s hvězdičkou *Open Configuration* a křížkem *Close Configuration* slouží k navázání a ukončení komunikace se serverem. Funkce *Open tag* se připojí k vypsaným tagům s nastaveným datovým typem. Jestliže tagy na serveru nejsou vytvořeny, pak je funkce na serveru vytvoří. Funkcí *Close tag* jsou tagy opět uzavřeny. Tyto funkce tagů se vážou k datovému typu tagu, který je definován. V jiném případě nebude tag fungovat správně. V Cloudu jsou vypsány v seznamu jednotlivé tagy s jejich aktuálními hodnotami. Pět tagů jsou stejného datového typu Double, proto jsou řešeny způsobem využívající pole stringů s názvy tagů a cyklu *For* pro optimalizování počtu funkcí a rozsáhlosti kódu. Toto řešení přiřazuje tagům v pevnou pozici při čtení nebo zápisu dat a musí na to být v dalším zpracování dat brán zřetel. Poslední tag by mohl být řešen stejným způsobem, protože je jediný jiného datového typu, má vlastní propojení s programem a nemá tedy žádnou pozici.

Tabulka 5.1: Seznam tagů

Pozice	Název tagu	Datový typ	Vstup/Výstup	Terminál
0	RC.Circuit	Double	výstup	OUTPUT
1	RC.Input	Double	výstup	OUTPUT
2	WEB.Rezistor	Double	vstup	Rezistor
3	WEB.Kondenzator	Double	vstup	Kondenzátor
4	WEB.Napeti	Double	vstup	Napětí
	WEB.Power	Bool	vstup	ON/OFF

Kapitola 6

Backend aplikace

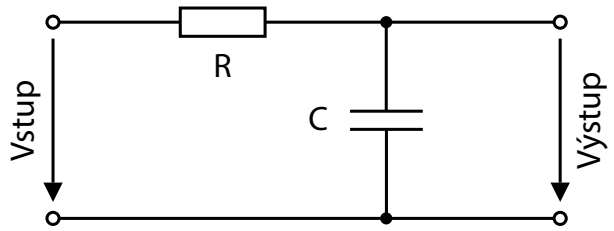
Backend aplikace běží na lokálním počítači umístěném v místě měření. Tato aplikace má za úkol měřit požadované měření, na které je backend aplikace naprogramována. Jak už název napovídá, aplikace běží na pozadí a jejím účelem není uživateli poskytovat uživatelské prostředí. To je uživateli zprostředkováno skrze frontend aplikaci, které se bude věnovat následující kapitola.

6.1 Data

Před vytvořením aplikace bylo třeba vybrat mezi fyzickým měřením s pomocí měřicích karet nebo simulovat nějaká data. V práci byla zvolena simulace přechodového jevu sériového RC obvodu. Tento jev byl následně naprogramován ve standardním vývojovém prostředí LabVIEW NXG. Za účelem aplikace nepřetržitě přenášet a zobrazovat data byl vytvořen cyklický program. Tedy periodicky se střídající průběhy vstupního napětí obvodu a nabíjení/vybíjení kondenzátoru. Pro získání požadovaných data pro zobrazení ve webové aplikaci, bylo použito vzorců pro nabíjení (6.1) a vybíjení kondenzátoru (6.2) pro konkrétní zapojení RC obvodu. [24, 25]

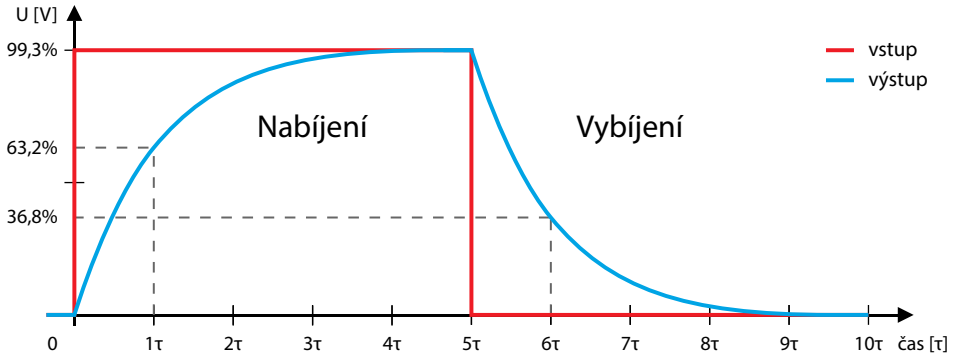
$$u_c = U_v \left(1 - e^{-\frac{t}{\tau}}\right) \tag{6.1}$$

$$u_c = U_v e^{-\frac{t}{\tau}} \tag{6.2}$$



Obrázek 6.1: Schéma RC obvodu

Jakmile je na vstup RC obvodu přivedeno vstupní napětí, kondenzátor je nabíjen přes rezistor v reakci na impuls. Vstupním napětím této simulace rozumíme ideální pravoúhlý impuls, tzn. okamžitá náběžná i sestupná hrana impulsu. Výstupní napětí by v ideálním případě přesně odpovídalo průběhu vstupního napětí. V praxi tomu však není. Napětí na kondenzátoru se nemůže okamžitě změnit, ale exponenciálně narůstá v závislosti časové konstantě τ. Tato časová konstanta odpovídá 63,2% nabíjení nebo 36,8% při vybíjení kondenzátoru. Kondenzátor se považuje za nabitý po uplynutí 5τ, což odpovídá nabití na 99,3%. Výchozí nastavení programu uvažuje právě s touto hodnotou nabití.

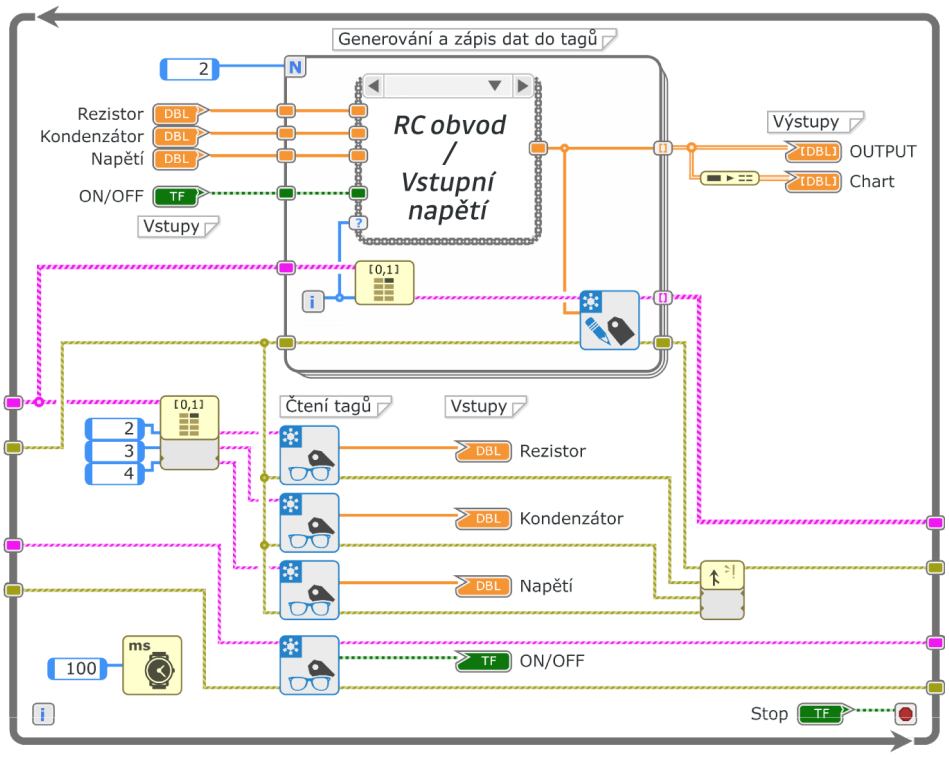


Obrázek 6.2: Časové průběhy kondenzátoru v RC obvodu

6.2 Hlavní program

Hlavní program aplikace zprostředkovává generování dat, popřípadě měření na fyzických přístrojích. Program je napojen pomocí API klíče a adresy SystemLink Cloudu na tagy obsahující data. V tomto případě jsou vstupní data zadávána ve webové aplikaci. Data jsou načítána pomocí SystemLink funkcí *Read Tag* a zapisována do vstupních terminálů aplikace. Aby byla vyčtena správná data do korespondujících terminálů, mají tagy předem určenou pozici (viz. tabulka 5.1). Díky tomu lze přesně určit, z kterého tagu jsou data čtena a do kterého zase zapisována. Vstupní data vstupují do cyklu *For*, ve kterém se nachází řídící struktura *Case* obdobná *Case and Switch* v jazyce C. V této struktuře se nachází dvě možnosti, každý obsahuje jeden podprogram generující simulační data. V závislosti na aktuální iteraci cyklu *For* jsou provedeny postupně oba podprogramy a zápis SystemLink funkcí *Write Tag* do příslušného tagu v rámci jednoho cyklu hlavního programu. V každém cyklu se tedy provede výčet všech vstupních dat a zápis výstupních dat do tagů. Pro kontrolu dat je čelní panel aplikace vizuálně podobný webové aplikaci a zobrazuje stejná data.

Aplikace je navržena tak, aby mohla být po menších úpravách týkajících se vstupních dat, vypuštěním simulačních částí a úpravou tagů nasazena na měření fyzickými přístroji.



Obrázek 6.3: Blokový diagram halvního programu

6.3 SubVI

Celá aplikace by mohla být napsána v jednom programu. Ten by však zabíral spoustu místa a stal by se nepřehledným. Simulace dat přechodového jevu RC obvodu je rozdělena do dvou samostatných podprogramů (SubVI), které by šlo použít i v jiných programech. SubVI jsou spouštěna v hlavním programu, jedno má na starosti simulaci nabíjení/vybíjení kondenzátoru v RC obvodu a druhé vstupní napětí obvodu. Jsou navrženy tak, aby při každém zavolání hlavním programem, vypsaly jednu hodnotu daného průběhu na výstup. Chod smyček v SubVI závisí na hlavním programu, je-li zavoláno konkrétní SubVI, nezávisle na druhém se provede v něm pouze jeden cyklus. Tím, že je každé SubVI voláno jednou během jednoho cyklu hlavního programu je zajištěn přísun pouze jedné výstupní hodnoty z každého SubVI.

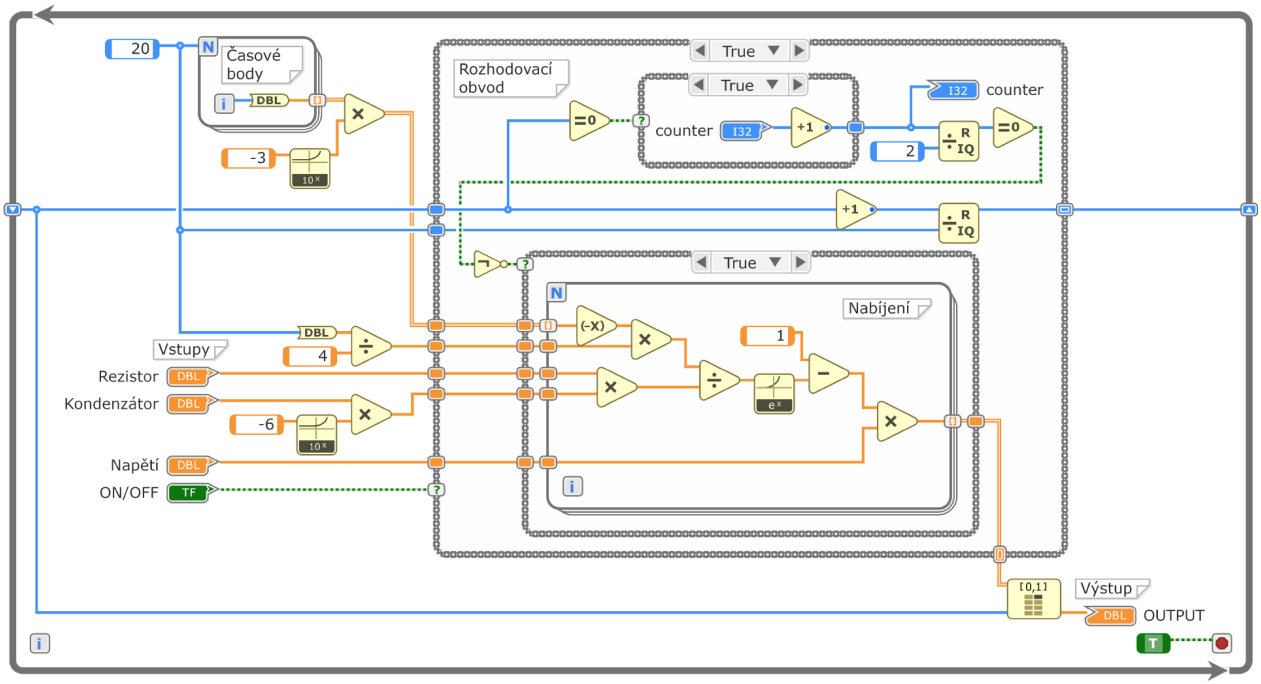
6.3.1 RC obvod

Byl vytvořen simulační program, který je schopen generovat hodnoty výstupního napětí na kondenzátoru. Pro účel aplikace byla nutná modifikace vzorců. Tato modifikace se týká času t, který nahrazují jednotlivé body nabývající dvaceti hodnot od 0 po 19 v množině přirozených čísel včetně nuly. V těchto jednotlivých bodech jsou následně vypočteny hodnoty napětí na kondenzátoru. Z počátku bylo generováno sto hodnot výstupního napětí. Simulace dat však byla zbytečně dlouhá.

Dosažení stejných výsledků s použitím menšího počtu bodů, bylo zapotřebí tyto body zvětšit na hodnoty původní křivky o sto bodech. Zvoleno byla pětinu bodů z původních sto, tedy dvacet bodů křivky. Výsledné body křivky o menším počtu bodů byly následně pět krát zvětšeny. Tato operace reprezentuje konstantu t v původním vzorci. Tyto hodnoty jsou následně děleny časovou konstantou τ, která se vypočte násobením hodnoty rezistoru a kondenzátoru. Nejnižší hodnotu, kterou může exponent nabývat jev našem případě nula. Když má Eulerovo číslo e v exponentu právě nulu, rovná se jedničce. Z toho důvodu je v čase nula hodnota napětí na kondenzátoru při nabíjení rovna nule (6.3). Naopak při jeho vybíjení je hodnota rovna vstupnímu napětí U_v (6.4).

$$u_c = U_v \left(1 - e^{-\frac{5x \cdot 10^{-3}}{RC}}\right), \quad x = \langle 0, 19 \rangle \subset \mathbb{N}_0 \tag{6.3}$$

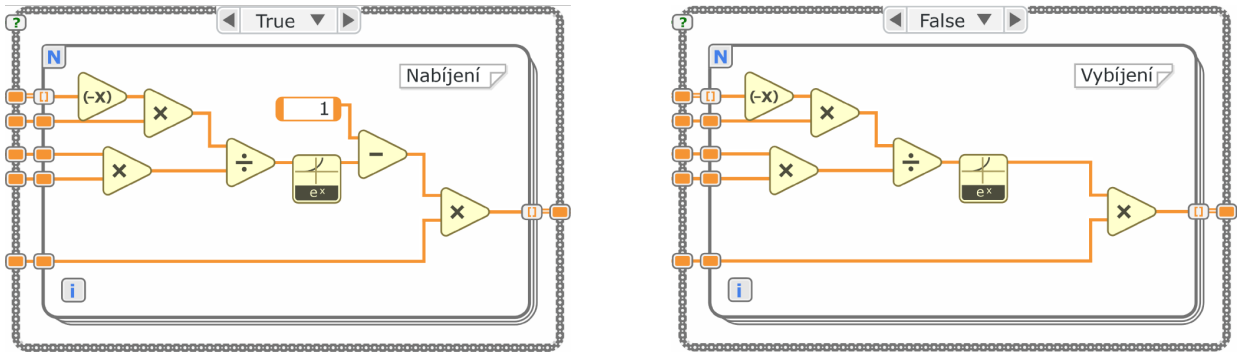
$$u_c = U_v e^{-\frac{5x \cdot 10^{-3}}{RC}}, \quad x = \langle 0, 19 \rangle \subset \mathbb{N}_0 \tag{6.4}$$



Obrázek 6.4: Blokový diagram SubVI generování výstupního napětí

Program v postupně pravidelných intervalech střídá jednotlivé křivky nabíjení a vybíjení kondenzátoru. Tohoto střídání je dosaženo za použití rozhodovacího obvodu, jež pracuje na principu rozpoznání sudého či lichého čísla. Pro získání tohoto čísla musel být vytvořen čítač, který přičítá pouze tehdy, když jsou vypsány všechny body dané křivky. Zde se opět dostáváme k číslu 20, tedy počtu bodů každé křivky. Vypisování jednotlivých bodů křivky a čítání čítače spolu úzce souvisí. Tyto operace jsou řešeny stejným posuvným registrem, který při každém cyklu programu přičte jed-

ničku do dalšího cyklu tzn. na vykreslení všech bodů křivky a instrukci pro čítač je zapotřebí dvaceti cyklů programu. Posuvný registr začíná hodnotou 0, ta se v posuvném registru objeví po dalších 19 cyklech programu. Čítač je inkrementován právě tehdy, kdy se v posuvném registru nachází nula. V závislosti na tomto čísle bude vykreslovaná křivka nabíjení nebo vybíjení.



(a) nabíjení kondenzátoru

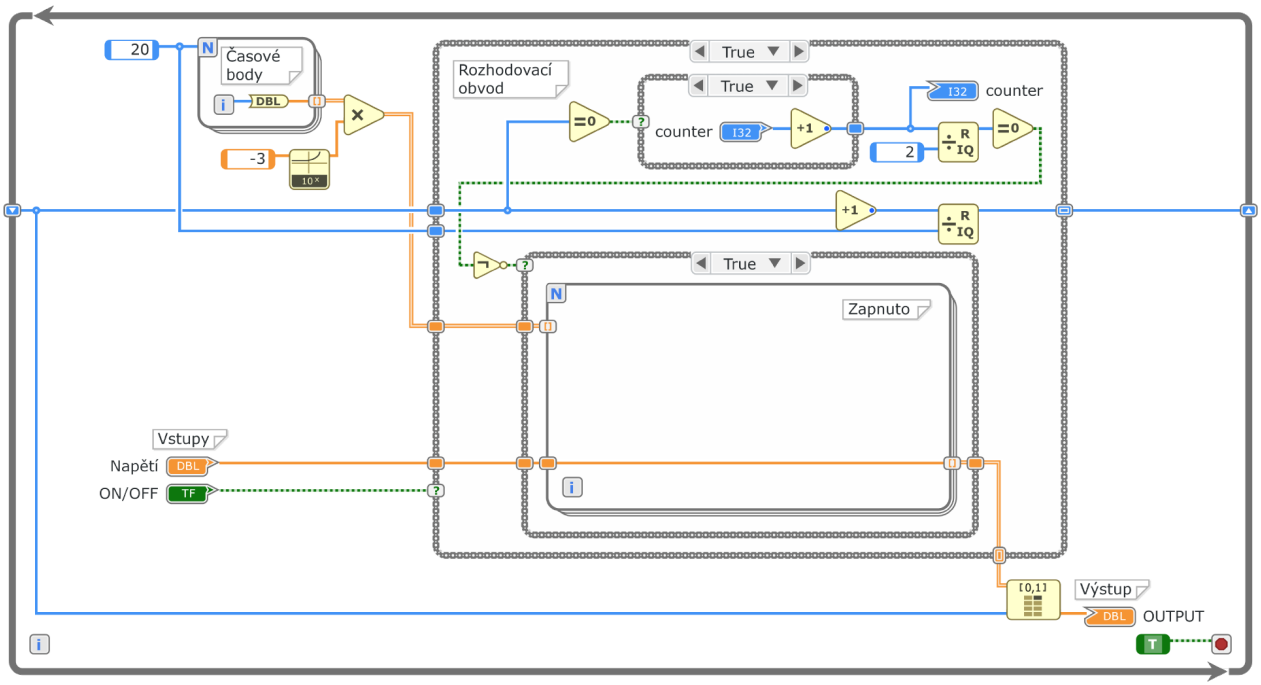
(b) vybíjení kondenzátoru

Obrázek 6.5: Výpočet dat výstupního napětí

Generování dat závisí především na vstupních datech přicházejících z hlavního programu. Při aktivaci tlačítka ON/OFF je spuštěna simulace začínající křivkou nabíjení kondenzátoru. Je tomu tak díky hodnotě čítače, který je po zapnutí simulace ve stavu 0. Jakmile uživatel stejným tlačítkem ON/OFF simulaci přeruší, čítač a posuvný registr jsou následně vynulovány tzn. na výstupní terminál jsou vypisovány nulové hodnoty.

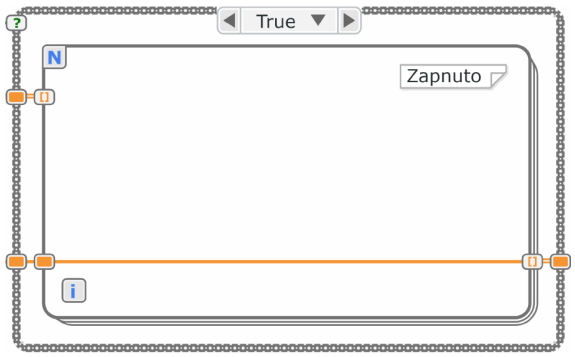
6.3.2 Vstupní napětí

SubVI generování vstupního napětí je řešeno obdobným způsobem, jako je tomu u dat RC obvodu. Stejným tlačítkem ON/OFF je spuštěna simulace začínající zapnutím vstupního napětí. Obvod rozhodování a chování čítače a je totožné jako s předchozím SubVI, tzn. čítač začíná v nule a čítá po dvaceti cyklech programu. Při generování hodnot vstupního napětí však není potřeba využívat žádného vzorce. Nejsou zde potřeba terminály pro parametry obvodu, kromě terminálu vstupního napětí. Z původního vzorce zbylo pouze generování časových bodů t. Je tomu tak z důvodu synchronizace generování, správná synchronizace vyžaduje stejný počet bodů pro obě SubVI. Pokud by obě SubVI neměly stejné časové body, čítač by nečítal ve stejnou chvíli. Ve výsledku by došlo k posunu průběhu jednoho nebo druhého SubVI při každé změně čítače.

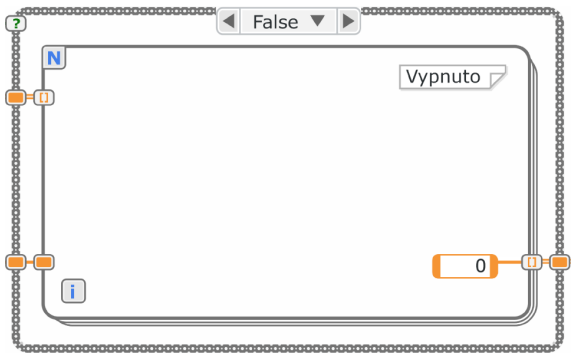


Obrázek 6.6: Blokový diagram SubVI generování vstupního napětí

Po zapnutí je spuštěna simulace a na výstupní terminál je přímo vypisována hodnota aktuálního napětí nastaveného na vstupním terminálu. Jakmile je simulace vypnuta, na výstupní terminál je posílána nulová hodnota.



(a) vstupní napětí zapnuto



(b) vstupní napětí vypnuto

Obrázek 6.7: Výpočet dat vstupního napětí

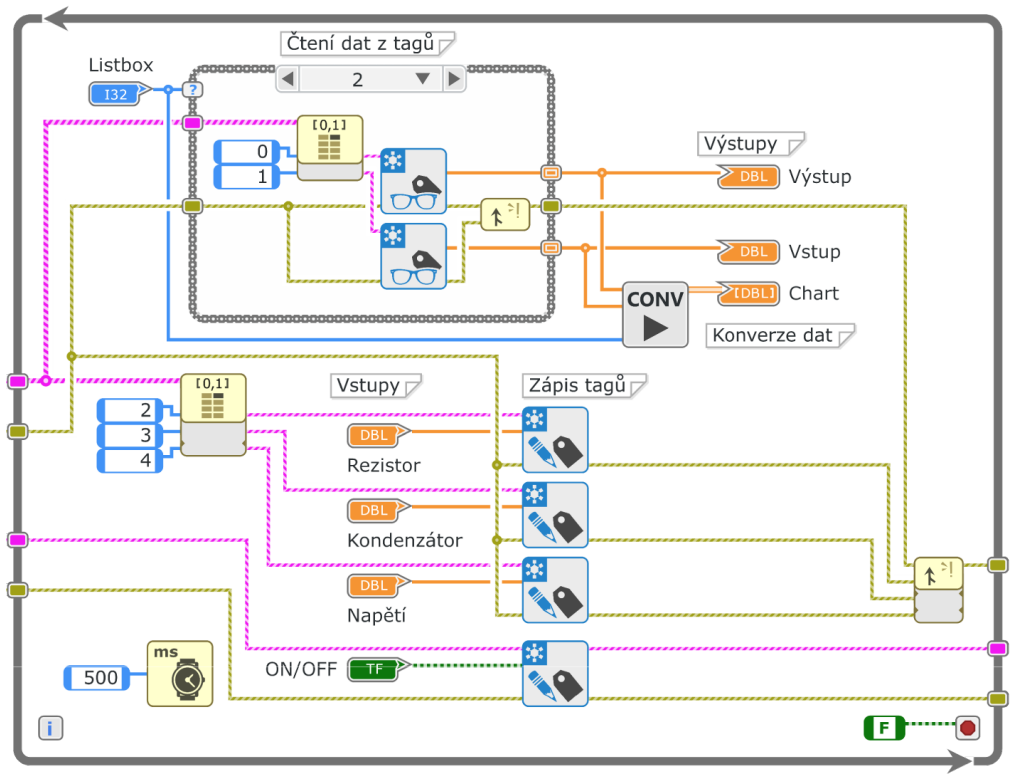
Kapitola 7

Frontend aplikace

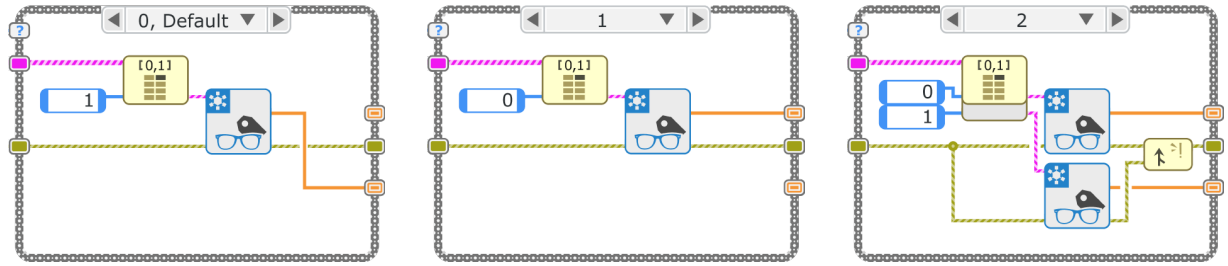
Webová aplikace je frontendem měřicí backend aplikace běžící na pozadí. Jedná se webové rozhraní tvořící vstupní a výstupní interface, umožňující uživateli aplikaci ovládat, zadávat vstupní parametry a zobrazovat data. Uživatel k webové aplikaci přistupuje skrze webový prohlížeč. Frontend aplikace je HTML stránka stylizovaná kaskádními styly CSS a doplněna JavaScript prvky, proto musí být z důvodu správného překladu ve webovém prohlížeči vytvořena ve WebVI, které právě toto umožňuje. Čelní panel aplikace je automaticky generován do HTML a CSS, zatímco blokový diagram je generován do JavaScriptu.

7.1 Blokový diagram

Z datového úložiště SystemLink Cloudu jsou načteny jednotlivé tagy vstupující do aplikace. Aplikace je navržena tak, aby bylo možné v aplikaci nastavovat vstupní parametry měření včetně zapínání simulace. Celá aplikace je naprogramována v nekonečné smyčce *While* tzn. dokud je aplikace spuštěná, nevypne se, pokud nenastane chyba programu. Jeden cyklus programu se provádí každých 500 ms. Zápis vstupních parametrů probíhá výběrem jednotlivých tagů. U tagů datového typu DBL je nutné zajistit správný výběr tagu podle jejich pozic v poli (viz. tabulka 5.1). Pomocí funkce *Index Array* je příslušný tag vybrán vložením čísla jeho pozice. U posledního tagu datového typu Bool výběr nemusí být zapotřebí a je naveden přímo na příslušný tag. Data přicházející z backend aplikace jsou vedena do řídící struktury *Case* obsahující tři možnosti. V této struktuře je prováděno čtení dat z tagů obsahujících výstupní data z měřicí aplikace. *Listbox* je ovládací prvek obsahující tři položky, každá položka má jinou hodnotu. Číslování těchto tří položek začíná nulou a končí dvojkou. V závislosti na aktuální volbě položky je vybrána příslušná možnost v *Case*. Na základě vyčtených dat, jsou dále zobrazována ve výstupních indikačních prvcích. Následné zobrazení dat v grafu vyžaduje převod přiváděných dat.



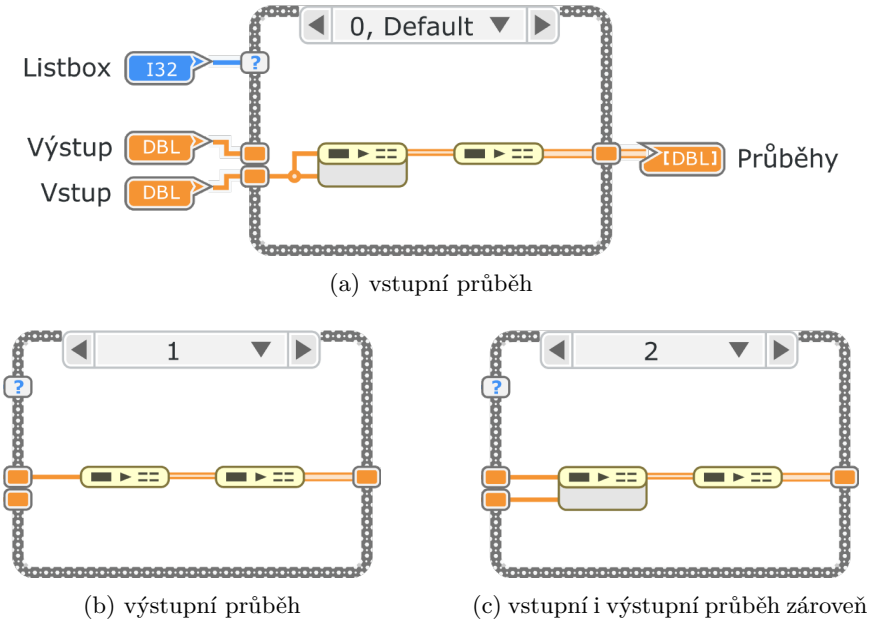
Obrázek 7.1: Blokový diagram webové aplikace



(a) tag RC.Input (b) tag RC.Circuit (c) tagy RC.Input a RC.Circuit

Obrázek 7.2: Možnosti čtení dat z tagů

Zobrazovač průběhů *Chart* není schopen zobrazovat data, která přicházejí z datových tagů přímo. Tato data jsou datového typu Double (DBL), které zobrazovač není chopen zobrazit, pokud nejsou ve správném formátu. Převod na správnou podobu dat zajišťuje toto webové SubVI. V něm je proveden převod pro jednotlivé možnosti zobrazení přes teminál *Listbox*. Přicházející DBL data jsou první funkcí *Build Array* převedeny do formátu 1D pole, druhou *Build Array* funkcí je toto pole převedeno na 2D pole, které je schopen zobrazovač průběhů zobrazit.



(a) vstupní průběh

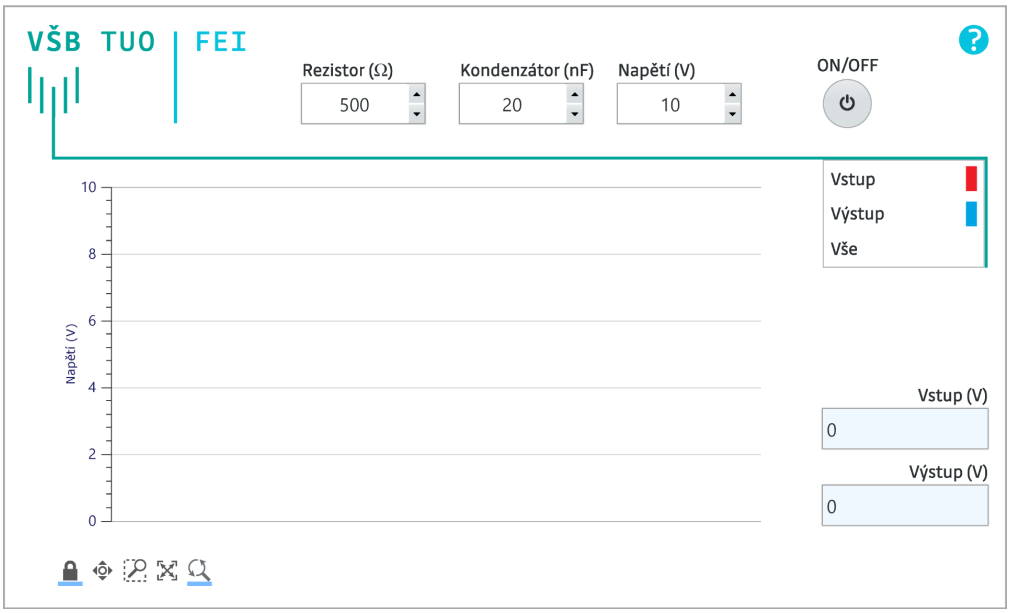
(b) výstupní průběh

(c) vstupní i výstupní průběh zároveň

Obrázek 7.3: Blokový diagram SubVI převodu dat a jeho možnosti

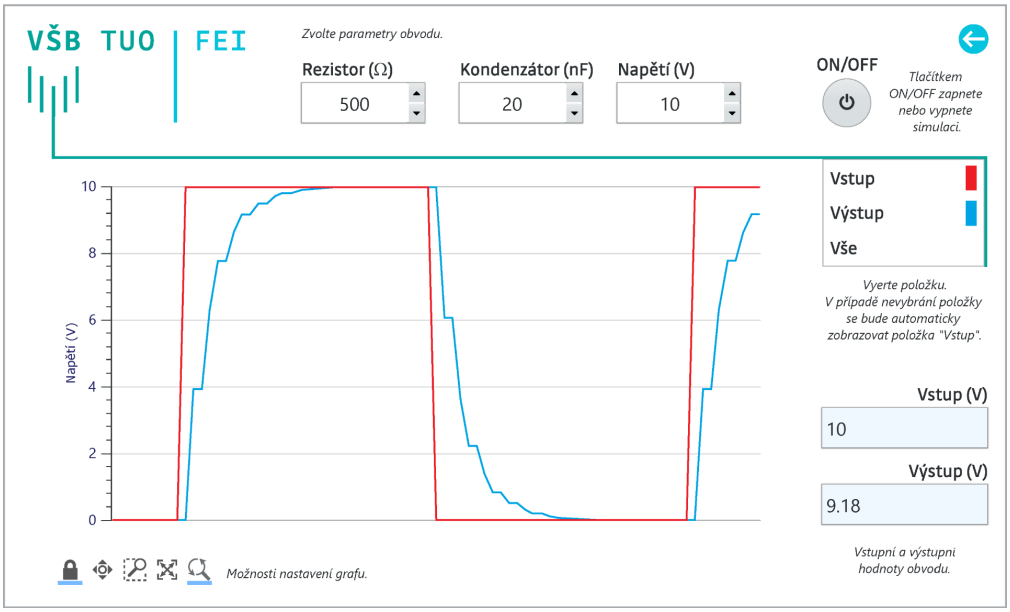
7.2 Čelní panel

Front panel webové aplikace je nejdůležitější součástí aplikace. Tato část aplikace je uživateli zobrazována v jeho prohlížeči a je jedinou, kterou běžný uživatel uvidí. Uživatel je tedy schopen vše nastavit a obsloužit ve webové aplikaci. K tomu slouží vstupní ovládací prvky hodnot *Rezistoru* (Ω), *Kondenzátoru* (nF) a vstupního *Napětí* (V). Tyto parametry jsou nutné pro správný běh simulace dat, proto jsou v prvcích nastaveny výchozí hodnoty, jež se nastaví při spuštění aplikace. Simulace dat je zahájena stiskem tlačítka ON/OFF. Výstupní data simulace jsou zobrazována číselně ve výstupních indikačních prvcích *Vstup* (V) a *Výstup* (V) v pravé dolní části aplikace. Stejná data jsou ve stejnou chvíli vynášena v grafu. Pod tlačítkem ON/OFF se nachází výběr z možností, které uživateli umožňují zobrazit požadovaný průběh dat. Tento výběrový prvek obsahuje tři možnosti: zobrazení vstupního průběhu, zobrazení výstupního průběhu nebo zobrazení vstupního a výstupního průběhu současně. Pokud uživatel nevybere žádnou z možností a simulace je spuštěna, zobrazuje se v grafu vstupní průběh obvodu. V levé dolní části aplikace se nachází nastavení grafu. Uživatel je pomocí těchto nástrojů schopen měnit přiblížení a oddálení zobrazení, množství zobrazených bodů v okně grafu a možnost prohlížet zpětně průběhy do vzdálenosti 512 bodů.



Obrázek 7.4: Hlavní stránka (Main)

Aplikace obsahuje dvě HTML stránky: hlavní *Main* a stránku nápovědy *Help*. V pravém horním rohu na hlavní stránce se nachází tlačítko nápovědy ‘ ? ’ s hypertextovým odkazem. Po jeho stisknutí je uživatel přesměrován na stránku s popisem aplikace s možností návratu zpět do aplikace tlačítkem ‘ ← ’. Při návratu zpět na hlavní stránku jsou parametry aplikace uvedeny zpět do výchozích přednastavených hodnot.



Obrázek 7.5: Stránka nápovědy (Help)

Na čelním panelu se nachází osm prvků sloužících pro ovládání nebo zobrazování dat. Každý z těchto prvků může být modifikován v nastavení prvku v LabVIEW, ale může být také upraven pomocí kaskádových stylů CSS. Logo školy, ikona nápovědy a tlačítka zpět jsou vložené soubory ve vektorovém formátu *.svg* vhodném pro webové stránky. Tyto soubory jsou velice malé a nezávislé na výsledném rozlišení tzn. při přiblížení stránky nedojde k jejich rozmazání. Jsou zde vloženy také prvky textových polí, které lze pomocí CSS obarvit a použít jako grafický prvek. Tyto prvky jsou použity jako barevné označení průběhů ve výběru zobrazení a zelená horizontální linka oddělující kontrolní část od zobrazovací části aplikace.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
      <title>Main</title>
      <style>
        ni-front-panel {
          background-color: White;
          box-shadow: 0px 0px 15px 3px rgba(0, 0, 0, 0.75);}
        .Section-Green {
          background-color: #00A499;}
        jqx-numeric-text-box {
          --ni-border: none;
          --ni-control-background-color: white;
          box-shadow: 0px 0px 1px 1px rgba(0, 0, 0, 0.5);}
      </style>
    <body>
      <div class="ni-front-panel-wrapper" vi-ref="" ni-control-id="FrontPanelWrapper">
        <ni-front-panel control-resize-mode='fixed' layout='absolute' ni-control-id='2'>
          <jqx-numeric-text-box binding-info='{"accessMode": "readOnly", "dataItem": "
              dataItem_Kondenzátor", "dco": 5, "isLatched": false, "prop": "value", "sync": false, "
              unplacedOrDisabled": false}' control-resize-mode='fixed' enable-mouse-wheel-action='true
              ' follower-ids='["9"]' label-alignment='top-left' label-id='9' max='Infinity' min='-Infinity'
              ni-control-id='6' ni-type='Double' radix='decimal' significant-digits='6' spin-buttons='
              true' spin-buttons-initial-delay='500' spin-buttons-position='right' spin-buttons-step='1'
               validation='interaction' value='20'></jqx-numeric-text-box>
          <ni-label control-resize-mode='fixed' label-alignment='top-left' ni-control-id='9' text='
              Kondenzátor (nF)'></ni-label>
          <ni-text class='Section-Green' control-resize-mode='fixed' ni-control-id='14' text='Text'></
              ni-text>
        </ni-front-panel>
      </div>
    </body>
  </head>
</html>
```

Výpis 7.1: HTML kód aplikace s CSS (upravený)

Vzhledem k velikosti celého kódu, byl kód pro ukázku vybraných prvků, včetně kaskádových stylů CSS, náležitě upraven. Kompletní zdrojové kódy jsou obsaženy v přílohách B a C. Kaskádové styly jsou do kódu nahrávány ze souboru a nejsou vypsány přímo v HTML kódu, jak je uvedeno výše. Pro importování souboru *style.css* je zapotřebí použít příkazu *<link href="style.css" rel="stylesheet">*.

- **ni-front-panel**: Hlavní okno aplikace je nastaveno na pevnou velikost. Responzivní řešení okna neumožňuje přesné umístění prvků. Umožňuje nastavit např. barvu pozadí nebo vrhání stínu okna aplikace pomocí CSS.
- **.Section-Green**: Tímto parametrem se po přiřazení k prvku změní jeho barva pozadí na zelenou např. u textového pole.
- **jqx-numeric-text-box**: Ovládací prvky jsou nastaveny pro zadávání s možností měnit hodnoty i kolečkem myši. Jejich velikost je pevně stanovena včetně limitů a výchozích hodnot. CSS odstraňuje základní rámeček ovládacích prvků, mění barvu výplně na bílou a přidává vrhání stínu. Každý prvek má své vlastní ID číslo, kterým lze určit, který prvek bude ovlivněn. Pokud není řečeno, který prvek má být ovlivněn, budou ovlivněny všechny prvky.

Kapitola 8

Závěr

V rámci bakalářské práce jsem se seznámil s vývojovým prostředím LabVIEW NXG, které se od klasického LabVIEW v mnohém liší. Tyto rozdíly obou prostředí jsem prostudoval a využil jich při dalším postupu práce. V rámci LabVIEW NXG jsem se primárně zabýval pluginem web module, který umožňuje vytvářet webové aplikace. Při práci s pluginem ve WebVI jsem došel k poznatku, že mnohé z běžně používaných funkcí klasického prostředí VI nelze použít. Dále jsem se zabýval cloudovým výpočetním prostředím jako vhodnou volbou pro uložení dat a komunikaci s aplikacemi. Webová aplikace je tvořena generovaným HTML kódem. Kód webové aplikace lze rozšířit o další webové prvky, např. úpravu vzhledu pomocí kaskádových stylů CSS, kterou jsem v aplikaci použil.

Řešení problému bylo sestaveno ze tří provázaných funkčních bloků. Webové aplikace, simulační, případně měřicí aplikace a komunikačního rozhraní mezi aplikacemi. Simulační část programu slouží ke generování dat z měření RC obvodu, který je matematicky simulován uvnitř daného programu. Tato část systému je plně modulární tzn. do budoucna je jednoduše nahraditelná za reálnou měřicí jednotku, která by takto posílala skutečná data z měření do vizualizační části.

Samotná vizualizace a přenos dat mezi aplikacemi jsou realizovány pomocí tagů, které se používají k datovému přenosu mezi distribuovanými systémy. Jednotlivé tagy a webová aplikace byly vytvořeny a nahrány na SystemLink Cloud, tedy na NI cloudové úložiště. Jsou dostupné přes libovolné zařízení s webovým prohlížečem a přístupem k internetu. Přínosem tohoto řešení je, že uživatel webové aplikace je schopen spouštět měření, kontrolovat měřená data, anebo dokonce provádět konfiguraci celého měření.

Výsledná práce je plně funkční a poskytuje koncovému uživateli veškeré průmyslově potřebné informace. Tento typ distribuovaného systému považuji osobně za velmi přínosný. Do budoucna bych se mu chtěl dále věnovat, protože jeho využití považuji za potřebný krok k další průmyslové revoluci, tedy krokem k průmyslu 4.0.

Literatura

1. KALKMAN, Cor J. LabVIEW: A software system for data acquisition, data analysis, and instrument control. *Journal of Clinical Monitoring*. 1995-01, roč. 11, č. 1, s. 51–58. ISSN 1573-2614. Dostupné z DOI: 10.1007/BF01627421.
2. MICHALEC, Libor. 30 let historie Labview [online]. [B.r.] [cit. 2020-11-12]. Dostupné z: https://vyvoj.hw.cz/30-let-historie-labview.html.
3. KODOSKY, Jeffrey. LabVIEW. *Proc. ACM Program. Lang.* 2020-06, roč. 4, č. HOPL. Dostupné z DOI: 10.1145/3386328.
4. SAVARAM, Ravindra. LabVIEW Basics [online]. [B.r.] [cit. 2020-11-14]. Dostupné z: https://mindmajix.com/labview-basics.
5. WONG, William G. What's the Difference Between LabVIEW 2017 and LabVIEW NXG? *Electronic Design* [online]. [B.r.] [cit. 2020-11-22]. Dostupné z: https://cdn.baseplatform.io/files/base/ebm/electronicdesign/document/2019/04/electronicdesign_15618_labviewni.pdf.
6. SINGH, Vibhutesh Kumar. Comparing NI LABVIEW & C Programming language [online]. [B.r.] [cit. 2020-12-01]. Dostupné z: https://www.divilabs.com/2013/07/comparing-ni-labview-c-programming.html.
7. SAVARAM, Ravindra. Block Diagram in LabVIEW [online]. [B.r.] [cit. 2020-11-24]. Dostupné z: https://mindmajix.com/labview/block-diagram-in-labview.
8. NI, Corp. *LabVIEW User Manual*. Austin, Texas, USA, s. 2-1–2-4, 2003-04. Č. Part Number 320999E-01.
9. PEŠKA, Robert. LabView NXG na vlastní oči [online]. [B.r.] [cit. 2020-11-22]. Dostupné z: https://vyvoj.hw.cz/labview-nxg-na-vlastni-oci.html.
10. SYSTEMS, g. Using LabVIEW NXG Web VIs for Ubiquitous Data Access [online]. [B.r.] [cit. 2020-11-28]. Dostupné z: https://www.gsystems.com/hubfs/Solutions/G%20Systems%20LabVIEW%20NXG%20Web%20Final%20Version.pdf.

11. NI, Corp. *SystemLink Resources* [online]. 2020 [cit. 2020-12-05]. Dostupné z: https://www.ni.com/content/ni/locales/cs-cz/support/documentation/supplemental/17/getting-started-with-systemlink.html.

12. TEST-DYNAMICS. *What can you do with SystemLink?* [Online] [cit. 2020-12-05]. Dostupné z: http://testdynamics.co.za/news/002.html.

13. NI, Corp. *SystemLink™ Architecture* [online]. 2020 [cit. 2020-12-05]. Dostupné z: https://www.ni.com/cs-cz/support/documentation/supplemental/18/systemlink--architecture.html.

14. NI, Corp. *SystemLink Cloud FAQ* [online] [cit. 2020-12-07]. Dostupné z: https://www.systemlinkcloud.com/faq.

15. NI, Corp. *SystemLink™ Cloud: An Overview* [online] [cit. 2020-12-07]. Dostupné z: https://www.ni.com/cs-cz/innovations/white-papers/19/systemlink--cloud--an-overview.html.

16. *World Wide Web: Web Pages* [online] [cit. 2021-04-28]. Dostupné z: https://www.tutorialspoint.com/internet_technologies/web_pages.htm.

17. DOMANTAS, G. What is HTML? The Basics of Hypertext Markup Language Explained. *Hostinger* [online]. [B.r.] [cit. 2020-12-09]. Dostupné z: https://www.hostinger.com/tutorials/what-is-html.

18. KYRNIN, Jennifer. *Sams Teach Yourself: HTML5 Mobile Application*. Indianapolis: SAMS, c2011. ISBN 978-0-672-33440-5.

19. *April 22, 1993: Mosaic Browser Lights Up Web With Color, Creativity* [online] [cit. 2020-12-12]. Dostupné z: https://www.wired.com/2010/04/0422mosaic-web-browser/.

20. DUCKETT, Jon. *HTML & CSS: design and build websites*. Indianapolis: Wiley, c2011. ISBN 978-1-118-00818-8.

21. W3SCHOOLS. *HTML Introduction* [online] [cit. 2020-12-12]. Dostupné z: https://www.w3schools.com/html/html_intro.asp.

22. *Internet Web Programming: JavaScript* [online] [cit. 2021-04-28]. Dostupné z: https://www.tutorialspoint.com/internet_technologies/javascript.htm.

23. NI, Corp. *Call SystemLink Data Services* [online] [cit. 2021-04-27]. Dostupné z: https://ni.github.io/webvi-examples/CallSystemLinkDataServices/.

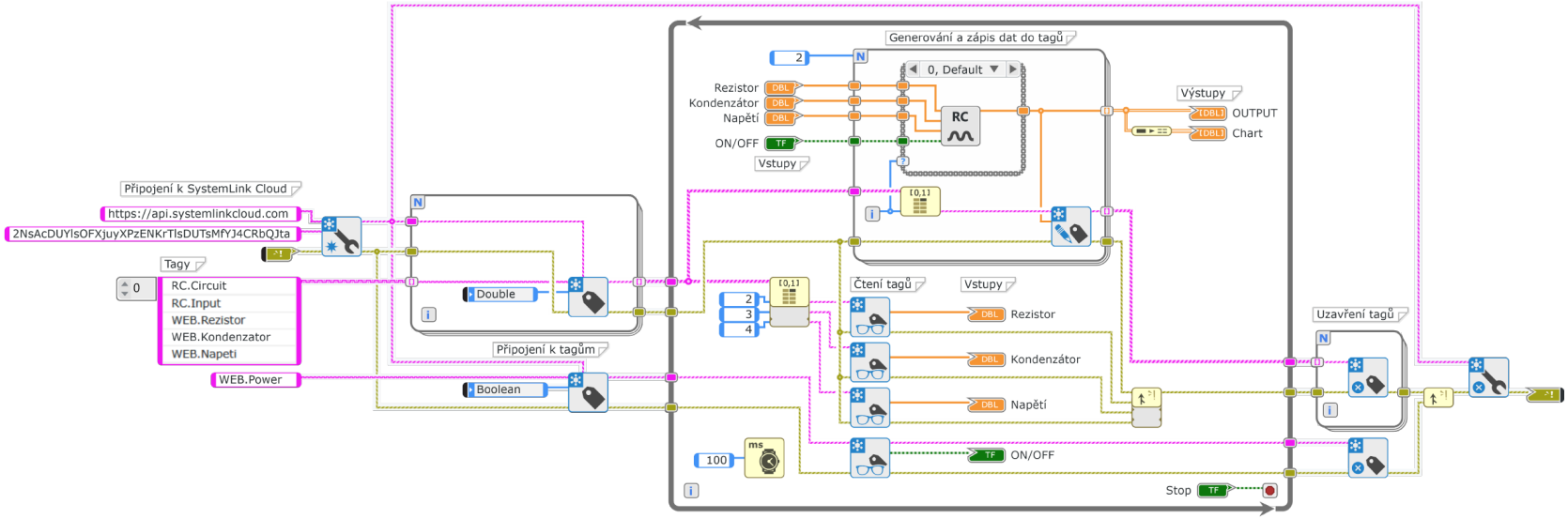
24. *RC Networks: RC Integrator* [online]. 2017 [cit. 2021-04-27]. Dostupné z: https://www.electronics-tutorials.ws/rc/rc-integrator.html.

25. *RC Networks: RC Waveforms* [online]. 2017 [cit. 2021-04-27]. Dostupné z: https://www.electronics-tutorials.ws/rc/rc_3.html.

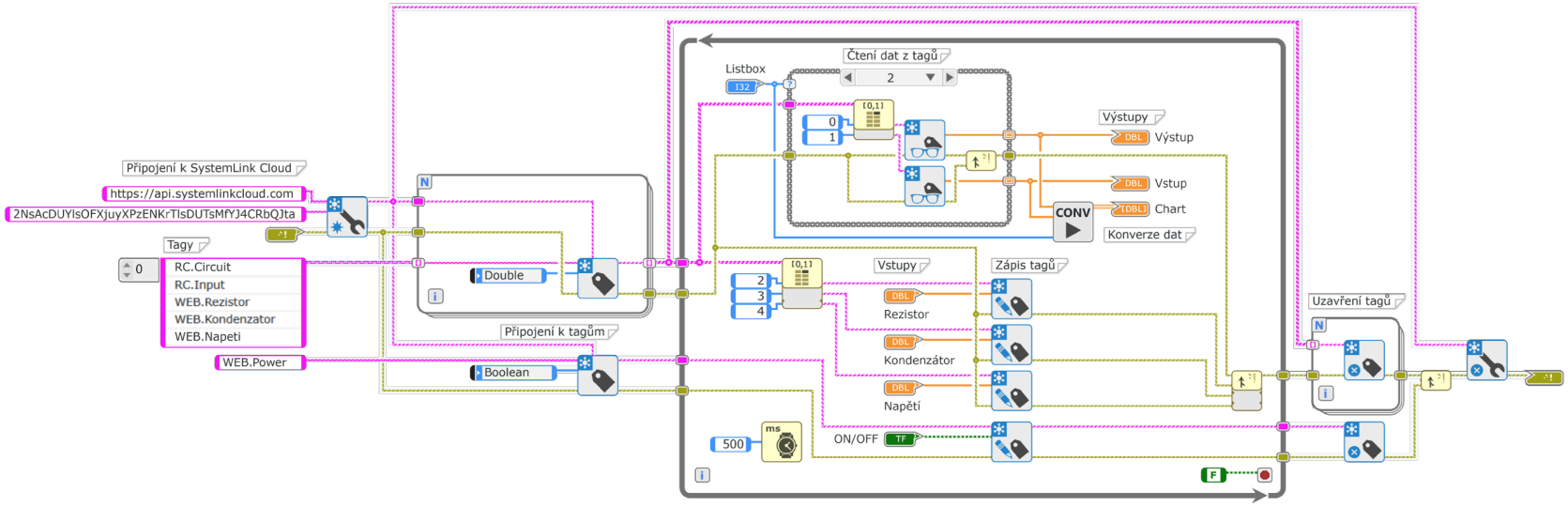
Seznam příloh

Příloha A

Blokové diagramy aplikací



Obrázek A.1: Měřící aplikace DataApp



Obrázek A.2: Webová aplikace WebApp

Příloha B

HTML webové aplikace

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
        <!---Read only section. Source panel edits to these document properties will not be persisted-->
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Main</title>
        <!---End read only section. Additional scripts or document properties can be added outside section-->
        <style ni-autogenerated-style-id="">
            /* Edits to the content of this style tag will not be persisted. To add additional styles, add a new
                style tag before or after this one */
        </style>
    <link href="style.css" rel="stylesheet">
</head>
<body>
    <div id="ni-outdated-browser-message" style="display:none">This browser is missing features needed to run
        this web application. Open this web application in a supported browser.</div>
    <div id="ni-failed-to-load-vireo-source" style="display:none">
        <div id="ni-failed-to-load-vireo-message-title">
            Unable to load resource file
        </div>
        <div id="ni-failed-to-load-vireo-message-body">
            Verify all WebVI files are available and served using an HTTP server or use a browser that can load
                files directly from the file system. See <a href="http://digital.ni.com/express.nsf/bycode/
                HOSTWEBAPP">Hosting a Web Application on a Server</a> for more information.
        </div>
    </div>
    <div class="ni-front-panel-wrapper" vi-ref="" ni-control-id="FrontPanelWrapper">
        <ni-front-panel control-resize-mode='fixed' layout='absolute' ni-control-id='2'>
            <jqx-list-box binding-info='{"accessMode": "readOnly", "dataItem": "dataItem_Listbox", "dco": 1, "
                isLatched": false, "prop": "selectedIndexes", "sync": false, "unplacedOrDisabled": false}' control
                -resize-mode='fixed' data-source='["Vstup", "Výstup", "Vše"]' follower-ids='["5"]' label-
```

alignment='top-left' label-id='5' ni-control-id='3' ni-type='Int32' selected-indexes='[]' selection-mode='one'></jqx-list-box>
<jqx-numeric-text-box binding-info='{"accessMode": "readOnly", "dataItem": "dataItem_Rezistor", "dco": 4, "isLatched": false, "prop": "value", "sync": false , "unplacedOrDisabled": false}' control-resize-mode='fixed' enable-mouse-wheel-action='true' follower-ids='["7"]' label-alignment='top-left' label-id='7' max='Infinity' min='-Infinity' ni-control-id='4' ni-type='Double' radix='decimal' significant-digits='6' spin-buttons='true' spin-buttons-initial-delay='500' spin-buttons-position='right' spin-buttons-step='1' validation='interaction' value='500'></jqx-numeric-text-box>
<jqx-numeric-text-box binding-info='{"accessMode": "readOnly", "dataItem": "dataItem_Kondenzátor", "dco": 5, "isLatched": false, "prop": "value", "sync": false, "unplacedOrDisabled": false}' control-resize-mode='fixed' enable-mouse-wheel-action='true' follower-ids='["9"]' label-alignment='top-left' label-id='9' max='Infinity' min='-Infinity' ni-control-id='6' ni-type='Double' radix='decimal' significant-digits='6' spin-buttons='true' spin-buttons-initial-delay='500' spin-buttons-position='right' spin-buttons-step='1' validation='interaction' value='20'></jqx-numeric-text-box>
<jqx-numeric-text-box binding-info='{"accessMode": "readOnly", "dataItem": "dataItem_Napětí", "dco": 6, "isLatched": false, "prop": "value", "sync": false, "unplacedOrDisabled": false}' control-resize-mode='fixed' enable-mouse-wheel-action='true' follower-ids='["11"]' label-alignment='top-left' label-id='11' max='Infinity' min='-Infinity' ni-control-id='8' ni-type='Double' radix='decimal' significant-digits='6' spin-buttons='true' spin-buttons-initial-delay='500' spin-buttons-position='right' spin-buttons-step='1' validation='interaction' value='10'></jqx-numeric-text-box>
<jqx-power-button binding-info='{"accessMode": "readOnly", "dataItem": "dataItem_ONOFF", "dco": 7, "isLatched": false, "prop": "value", "sync": false, "unplacedOrDisabled": false}' click-mode='release' control-resize-mode='fixed' false-content='' follower-ids='["13"]' label-alignment='top-center' label-id='13' ni-control-id='10' orientation='horizontal' switch-mode='click' true-content=''></jqx-power-button>
<ni-chart binding-info='{"accessMode": "writeOnly", "dataItem": "dataItem_Chart", "dco": 3, "isLatched": false, "prop": "value", "sync": true, "unplacedOrDisabled": false}' buffer-size='512' control-resize-mode='resize-horizontally' follower-ids='["16", "17", "18", "19", "15"]' label-alignment='top-left' label-id='15' metadata-overrides-plot-names='true' ni-control-id='12' ni-type='{"name": "Array", "rank": 2, "subtype": "Double"}' plot-area-margin='' read-only='true' value='{"data": [], "size": 512, "startIndex": 0, "valueType": "HistoryBuffer"}'>
<ni-cartesian-axis auto-scale='sliding-window' axis-position='bottom' format='LVRelativeSeconds:G6' grid-lines='true' label='Time' maximum='80' minimum='0' ni-control-id='20' show-label='true' show-minor-ticks='true' show-tick-labels='all' show-ticks='true' time-format-epoch='-2082844800000'></ni-cartesian-axis>
<ni-cartesian-axis auto-scale='exact' axis-position='left' format='LVRelativeSeconds:G3' grid-lines='true' label='Napětí (V)' maximum='10' minimum='0' ni-control-id='21' show='true' show-label='true' show-minor-ticks='true' show-tick-labels='all' show-ticks='true' time-format-epoch='-2082844800000'></ni-cartesian-axis>
<ni-cartesian-plot hover-format='{0}, {1}' label='Plot' ni-control-id='22' show='true' show-label='true' xaxis='20' yaxis='21'>
<ni-cartesian-plot-renderer area-base-line='negativeinfinity' area-fill='' bar-base-line='negativeinfinity' bar-fill='' line-stroke='rgba(0,164,229,1)' line-style='solid' line-width='2' ni-control-id='23' point-color='' point-shape='ellipse' point-size

```
        ='7.0710678118654755'></ni-cartesian-plot-renderer>
    </ni-cartesian-plot>
    <ni-cartesian-plot hover-format='{0}, {1}' label='Plot 2' ni-control-id='24' show='true' show
        -label='true' xaxis='20' yaxis='21'>
        <ni-cartesian-plot-renderer area-base-line='negativeinfinity' area-fill='' bar-base-line='
            negativeinfinity' bar-fill='' line-stroke='rgba(238,28,37,1)' line-style='solid' line-
            width='2' ni-control-id='25' point-color='' point-shape='ellipse' point-size
            ='7.0710678118654755'></ni-cartesian-plot-renderer>
    </ni-cartesian-plot>
</ni-chart>
<ni-label class='ni-hidden' control-resize-mode='fixed' label-alignment='top-left' ni-control-id
    ='5' text='Listbox'></ni-label>
<ni-label control-resize-mode='fixed' label-alignment='top-left' ni-control-id='7' text='Rezistor
    (Omega)'></ni-label>
<ni-label control-resize-mode='fixed' label-alignment='top-left' ni-control-id='9' text='Kondenz
    átor (nF)'></ni-label>
<ni-label control-resize-mode='fixed' label-alignment='top-left' ni-control-id='11' text='Napětí
    (V)'></ni-label>
<ni-label control-resize-mode='fixed' label-alignment='top-center' ni-control-id='13' text='ON/
    OFF'></ni-label>
<ni-label class='ni-hidden' control-resize-mode='fixed' label-alignment='top-left' ni-control-id
    ='15' text='Chart'></ni-label>
<ni-scale-legend class='ni-hidden' control-resize-mode='fixed' graph-ref='12' ni-control-id='16'
    ></ni-scale-legend>
<ni-plot-legend class='ni-hidden' control-resize-mode='fixed' graph-ref='12' ni-control-id='17'
    ></ni-plot-legend>
<ni-cursor-legend class='ni-hidden' control-resize-mode='fixed' graph-ref='12' ni-control-id
    ='18'></ni-cursor-legend>
<ni-graph-tools control-resize-mode='fixed' graph-ref='12' ni-control-id='19'></ni-graph-
    tools>
<ni-text class='Section-Green' control-resize-mode='fixed' ni-control-id='14' text='Text'></ni-
    text>
<ni-url-image alternate='' binding-info='{"accessMode": "writeOnly", "dataItem": "
    dataItem_URLImage", "dco": 8, "isLatched": false, "prop": "source", "sync": false , "
    unplacedOrDisabled": true}' control-resize-mode='fixed' follower-ids='["28"]' href='' label-
    alignment='top-left' label-id='28' ni-control-id='26' read-only='true' source='Images/FEI.
    svg' stretch='uniform' target='_self'></ni-url-image>
<ni-label class='ni-hidden' control-resize-mode='fixed' label-alignment='top-left' ni-control-id
    ='28' text='URL Image'></ni-label>
<ni-text class='Section-Green' control-resize-mode='fixed' ni-control-id='27' text='Text'></ni-
    text>
<ni-text class='Section-White' control-resize-mode='fixed' ni-control-id='29' text='Text'></ni-
    text>
<ni-text class='Section-White' control-resize-mode='fixed' ni-control-id='30' text='Text'></ni-
    text>
<ni-url-image alternate='' binding-info='{"accessMode": "writeOnly", "dataItem": "
    dataItem_URLImage_2", "dco": 9, "isLatched": false, "prop": "source", "sync": false, "
```

```
            unplacedOrDisabled": true}' control-resize-mode='fixed' follower-ids='["33"]' href='Help.html'
            label-alignment='top-left' label-id='33' ni-control-id='31' read-only='true' source='Images/
            HELP.svg' stretch='uniform' target='_self'></ni-url-image>
        <ni-label class='ni-hidden' control-resize-mode='fixed' label-alignment='top-left' ni-control-id
            ='33' text='URL Image_2'></ni-label>
        <jqx-numeric-text-box binding-info='{"accessMode": "writeOnly", "dataItem": "dataItem_Výstup",
            "dco": 0, "isLatched": false, "prop": "value", "sync": false, "unplacedOrDisabled": false}' control-
            resize-mode='fixed' enable-mouse-wheel-action='true' follower-ids='["35"]' label-alignment='
            top-right' label-id='35' max='Infinity' min='-Infinity' ni-control-id='32' ni-type='Double'
            radix='decimal' readonly='true' significant-digits='3' spin-buttons-initial-delay='500' spin-
            buttons-position='right' spin-buttons-step='1' validation='interaction' value='0'></jqx-
            numeric-text-box>
        <ni-label control-resize-mode='fixed' label-alignment='top-right' ni-control-id='35' text='Vý
            stup (V)'></ni-label>
        <jqx-numeric-text-box binding-info='{"accessMode": "writeOnly", "dataItem": "dataItem_Vstup", "
            dco": 2, "isLatched": false, "prop": "value", "sync": false, "unplacedOrDisabled": false}' control-
            resize-mode='fixed' enable-mouse-wheel-action='true' follower-ids='["37"]' label-alignment='
            top-right' label-id='37' max='Infinity' min='-Infinity' ni-control-id='34' ni-type='Double'
            radix='decimal' readonly='true' significant-digits='3' spin-buttons-initial-delay='500' spin-
            buttons-position='right' spin-buttons-step='1' validation='interaction' value='0'></jqx-
            numeric-text-box>
        <ni-label control-resize-mode='fixed' label-alignment='top-right' ni-control-id='37' text='Vstup
             (V)'></ni-label>
        <ni-text class='Section-White' control-resize-mode='fixed' ni-control-id='36' text='Text'></ni-
            text>
        <ni-text class='Section-LineRed' control-resize-mode='fixed' ni-control-id='38' text='Text'></
            ni-text>
        <ni-text class='Section-LineBlue' control-resize-mode='fixed' ni-control-id='39' text='Text'></
            ni-text>
    </ni-front-panel>
  </div>
</body>
</html>
```

Výpis B.1: Generovaný HTML kód webové aplikace

Příloha C

CSS webové aplikace

```
ni-front-panel {
    background-color: White;
    box-shadow: 0px 0px 15px 3px rgba(0, 0, 0, 0.75);}
.Section-Green {
    background-color: #00A499;}
.Section-Blue {
    background-color: #05C3DE;}
.Section-White {
    background-color: White;}
.Section-LineRed {
    background-color: #EE1C25;}
.Section-LineBlue {
    background-color: #00A4E5;}
ni-chart {
    background-color: white;
    padding-left: 10px;
    --ni-border: none;
    width: 100%;}
jqx-list-box {
    background-color: #00A499;
    padding-right: 3px;
    width: 100%;}
jqx-power-button {
    --ni-true-background: #05C3DE;}
jqx-numeric-text-box {
    --ni-border: none;
    --ni-control-background-color: white;
    --ni-indicator-background-color: AliceBlue;
    box-shadow: 0px 0px 1px 1px rgba(0, 0, 0, 0.5);}
```

Výpis C.1: Soubor CSS pro webovou aplikaci

Příloha D

Příloha v IS EDISON

- **DataApp**
  - DataApp.gvi — VI měřicí aplikace
  - DataApp.lvproject — projekt pro LabVIEW NXG
  - **SubVI**
    - IN.gvi — VI generování vstupního napětí
    - RC.gvi — VI generování výstupního napětí
- **WebApp**
  - WebApp.lvdist — balíček/instalátor pro LabVIEW NXG
  - WebApp.lvproject — projekt pro LabVIEW NXG
  - webapp_1.0.0.17_windows_x64.nipkg — balíček aplikace
  - **WebApp.gcomp**
    - CONV.gviweb — WebVI převodu dat
    - Help.gviweb — WebVI stránka nápovědy
    - Main.gviweb — WebVI hlavní stránka
    - style.css — kaskádové styly CSS
    - WebApp.gcomb — aplikace/knihovna pro LabVIEW NXG
    - **Images**
      - BACK.svg — obrázek tlačítka ‘ ← ’
      - FEI.svg — obrázek loga školy
      - GRAPH.svg — obrázek průběhů pro stránku nápovědy
      - HELP.svg — obrázek tlačítka ‘ ? ’